Κεφάλαιο 1ο: Βασικές Έννοιες

1.1 Συνήθεις Πράξεις

Το *Mathematica* υποστηρίζει όλες τις αριθμητικές πράξεις, και μάλιστα με τον γνωστό τρόπο. Έτσι μπορούμε, να προσθέσουμε δύο αριθμούς χρησιμοποιώντας το γνωστό σύμβολο "+",

```
100 + 200

300
```

να αφαιρέσουμε δύο αριθμούς χρησιμοποιώντας το γνωστό σύμβολο "-",

```
300 - 500

-200
```

να πολλαπλασιάσουμε δύο αριθμούς χρησιμοποιώντας το σύμβολο "*"

```
200  300

60000
```

και να διαιρέσουμε δύο αριθμούς χρησιμοποιώντας το σύμβολο "/".

```
1000 / 20

50
```

Το σύμβολο "*" που χρησιμοποιήσαμε για τον πολλαπλασιασμό μπορεί να αντικατασταθεί από το κενό διάστημα.

```
200 300

60000
```

Εκτός από το σύμβολο "/", το οποίο το χρησιμοποιήσαμε για τη διαίρεση αριθμών, μπορούμε να χρησιμοποιήσουμε και το συνδυασμό των πλήκτρων Ctrl + /. Για παράδειγμα, η επόμενη εντολή πληκτρολογήθηκε με την εξής σειρά: 1000 Ctrl + / 20.

```
1000
iiiiiiiii
  20

50
```

Όλες οι πράξεις που είδαμε παραπάνω μπορούν να συνδυαστούν μεταξύ τους.

```
12  10 + 24 − 12 ê 6

142
```

Πρέπει να σημειωθεί ότι υπάρχει συγκεκριμένη προτεραιότητα στην εκτέλεση των πράξεων. Συγκεκριμένα, 1η προτεραιότητα έχει η πράξη της ύψωσης σε δύναμη (^), 2η προτεραιότητα έχουν οι πράξεις του πολλαπλασιασμού (*) και της διαίρεσης (/) και 3η προτεραιότητα έχουν οι πράξεις της πρόσθεσης (+) και της αφαίρεσης (-). Μπορούμε να παρακάμψουμε όμως αυτή την προτεραιότητα εκτέλεσης των πράξεων χρησιμοποιώντας παρενθέσεις. Όταν χρησιμοποιούμε παρενθέσεις, **πρώτα εκτελούνται οι πράξεις μέσα στην παρένθεση,** και μετά οι πράξεις έξω από τις παρενθέσεις, πάντα με την προτεραιότητα εκτέλεσης των πράξεων που αναφέραμε πιο πάνω. Το παρακάτω παράδειγμα είναι το ίδιο με το προηγούμενο, με μόνη διαφορά τη χρήση παρενθέσεων. Το αποτέλεσμα ασφαλώς δεν είναι είναι το ίδιο.

```
12  10 + H24 − 12L ê 6

122
```

Παρατηρούμε ότι το *Mathematica* αριθμεί αυτόματα κάθε εισαγόμενη εντολή του χρήστη (Input), καθώς και την αντίστοιχη εξερχόεμνη απάντηση (Output) με έναν αριθμό. Η αρίθμηση αυτή είναι συνεχής, και αρχίζει από την αρχή σε κάθε επανεκίνηση του *Mathematica*. Η αρίθμηση αυτή είναι συνεχής, και αρχίζει από την αρχή σε κάθε επανεκκίνηση του *Mathematica*.

In[n]	Εντολή (Input) αριθμός n.
Out[n]	Απάντηση (Output) αριθμός n.

Το mathematica μας δίνει τη δυνατότητα να αναφερθούμε σε κάποιο συγκεκριμένο από τα προηγούμενα αποτελέσματα, χρησιμοποιώντας το σύμβολο " %n " όπου n είνα ο αριθμός του βήματος του αποτελέσματος που μαες ενδιαφέρει.Ειδικότερα,όταν θέλουμε να αναφερθούμε στο τελευταίο αποτέλεσμα,τότεχρησιμοποιούμε μόνο το σύμβολο " % ".Ο συμβολισμός " %% " μας δίνει το δεύτερο από το τέλος αποτέλεσμα,κ.τ.λ.

```
% - 22

100
```

Στην προηγούμενη εντολή αφαιρούμε από το τελευταίο αποτέλεσμα (12 * 10 + (24 - 12) / 6 = 122) τον αριθμό 20. Ενώ στην επόμενη εντολή διαιρούμε με το 200, το αποτέλεσμα του βήματος 5 (200 300 = 60000).

```
%5 ê 200

300
```

Μερικές φορές τα αριθμητικά αποτελέσματα, που δίνονται από το *Mathematica,* δεν είναι τα επιθυμητά. Για πράδειγμα, το πηλίκο της διαίρεσης

```
10000001 ê 101

 10000001
 ¯¯¯¯¯¯¯¯
   101
```

θα επιθυμούσαμε να δωθεί ως δεκαδικός αριθμός και όχι ως ένα πηλίκο που από την αρχή γνωρίζαμε. Το *Mathematica* προσπαθεί να δίνει την ακριβέστερη απάντηση που μπορεί, σε σχέση πάντα με τον τύπο των αριθμών που εισάγει ο χρήστης. Όταν, λοιπόν, ζητάμε να μας δώσει το πηλίκο δύο ακεραίων, το *Mathematica* προσπαθεί να επιστρέψει έναν ακέραιο. Επειδή όμως, το πηλίκο δεν είναι ακέραιος μας επιστρέφει την αμέσως μετά τον ακέραιο ακριβέστερη απάντηση, που φυσικά είναι ένας ρητός αριθμός.

Για να πάρουμε το πηλίκο της διαίρεσης 10000001/101 ως δεκαδικό αριθμό, αρκεί να εισάγουμε τον έναν από τους δύο ακεραίους με μορφή δεκαδικού (πραγματικού) αριθμού.

```
10000001.0 ê 101

99009.9
```

Το ίδιο αποτέλεσμα μπορούμε να το επιτύχουμε με τη χρήση της συνάρτησης N[expr], όπου expr είναι μια παράσταση. Η συνάρτηση N επιστρέφει την αριθμητική τιμή της παράστασης expr.

N[expr]	Επιστρέφει με προσέγγιση την αριθμητική τιμή της παράστασης expr.
N[expr, n]	Επιστρέφει με προσέγγιση την αριθμητική τιμή της παράστασης expr, χρησιμοποιώντας n ψηφία, στα οποία συμπεριλαμβάνεται και το ακέραιο μέρος της αριθμητικής τιμής.

```
N@10000001ê101D

99009.9
```

```
N@10000001ê101, 20D

99009.910891089108911
```

Δύο πολύ γνωστές σταθερές στο *Mathematica* είναι ο αριθμός π και ο αριθμός e.

Pi ή **p** Ο συμβολισμός του αριθμού π
E ή ‰ Ο συμβολισμός του αριθμου e
I ή Â Ο συμβολισμός του αριθμού è!!!!-1

```
N@Pi, 200D

3.14159265358979323846264338327950288419716939937510582097494¥
 59230781640628620899862803482534211706798214808651328230664709¥
 93844609550582231725359408128481117450284102701938521105559644¥
 46229489549303820
```

```
N@E, 200D

2.71828182845904523536028747135266249775724709369995957496696767¥
 62772407663035354759457138217852516642742746639193200305992181¥
 74135966290435729003342952605956307381323286279434907632338298¥
 98807531952510190
```

Στις δύο προηγούμενες εντολές, παρατηρούμε ότι το *Mathematica* χρησιμοποιεί στο τέλος κάθε γραμμής το σύμβολο με τις τρεις πλάγιες τελείες, για να δείξει ότι ο αριθμός συνεχίζεται και σε επόμενη γραμμή.

```
H2 + 3 IL H5 − 3 IL H−1 − 5 IL

26 − 104 ą
```

Με την ίδια ευκολία που κάνουμε πράξεις με ακεραίους και μιγαδικούς αριθμούς μπορούμε να κάνουμε και πράξεις με μιγαδικούς αριθμούς.

```
(2 + 3 I) / (4 + 5 I)
```

$$\frac{23}{41} + \frac{2}{41} i$$

```
(2.0 + 3.0 I) / (4.0 + 5.0 I)
```

```
0.560976 + 0.0487805 ⅈ
```

Το *Mathematica* διαθέτει συναρτήσεις με τις οποίες μπορούμε να διαχειριστούμε τους μιγαδικούς αριθμούς.

Re[z]	Δίνει το πραγματικό μέρος του μιγαδικού αριθμού z
Im[z]	Δίνει το φανταστικό μέρος του μιγαδικού αριθμού z
Abs[z]	Δίνει την απόλυτη τιμή ενός πραγματικού ή το μέτρο ενός μιγαδικού αριθμού z
Arg[z]	Δίνει την γωνία φ για την οποία ισχύει $z=\|z\|\cdot e^{i\varphi}$
Sqrt[a]	Δίνει την τετραγωνική ρίζα του αριθμού α

```
Re@(2 + 3 I) (5 - 3 I) (-1 - 5 I)D
```

```
26
```

```
Im@(2 + 3 I) (5 - 3 I) (-1 - 5 I)D
```

```
-104
```

```
Abs@(2 + 3 I) (5 - 3 I) (-1 - 5 I)D
```

$$26\sqrt{17}$$

```
Arg@(2 + 3 I) (5 - 3 I) (-1 - 5 I)D
```

```
-ArcTan@4D
```

```
Sqrt@49D

7
```

Πρέπει να σημειωθεί ότι το *Mathematica* δεν εκτελεί μόνο αριθμητικές πράξεις, αλλά περιέχει εκατοντάδες ενσωματωμένες συναρτήσεις, οι οποίες είτε μόνες τους είτε σε συνδυασμό μεταξύ τους επιλύουν διάφορα προβλήματα. Συναρτήσεις λέγονται οι διάφορες εντολές που χρησιμοποιούνται στο *Mathematica*. Η συνάρτηση **N** που είδαμε παραπάνω είναι μία από αυτές. Η μορφή που έχουν οι συναρτήσεις του *Mathematica* είναι η εξής:

FunctionName[ ]

όπου FunctionName είναι το όνομα της συνάρτησης. Συνήθως τα ονόματα των συναρτήσεων είναι ολόκληρες λέξεις, και έχουν σχέση με το αποτέλεσμα που θα προκύψει από την εκτέλεση της συνάρτησης, π.χ. Factor, Integrate. Το πρώτο γράμμα του ονόματος είναι πάντα κεφαλαίο. Όταν το όνομα μιας συνάρτησης αποτελείται από δύο ή περισσότερες λέξεις, τότε το πρώτο γράμμα κάθε λέξης γράφεται πάντα κεφαλαίο π.χ. FullForm, FactorInteger. Μεταξύ των αγκυλών τοποθετούμε το όρισμα ή τα ορίσματα της συνάρτησης. Πολλές συναρτήσεις, εκτός των ορισμάτων, μπορούν να δεχθούν και διάφορες επιλογές, οι οποίες τοποθετούνται επίσης μεταξύ των αγκύλων. Οι επιλογές αυτές είναι οδηγίες, τις οποίες ακολουθεί η συνάρτηση, για να παρουσιάσει το αποτέλεσμα με τον τρόπο που θέλουμε.

Ανεξάρτητα από αυτό που πληκτρολογεί ο χρήστης, το *Mathematica* βλέπει μόνο συναρτήσεις. Για παράδειγμα, όταν θέλουμε να προσθέσουμε δύο αριθμούς, εμείς φυσικά χρησιμοποιούμε το σύμβολο της πρόσθεσης, όμως το *Mathematica* βλέπει τη συνάρτηση Plus. Για να δούμε αυτό που βλέπει το *Mathematica* και όχι αυτό που βλέπουμε εμείς στο παράθυρο εργασίας, χρησιμμοποιούμε τη συνάρτηση **FullForm**.

FullForm[expr] Δίνει την πλήρη μορφή της παράστασης expr

Έτσι, η πρώτη από τις επόμενες εντολές προσθέτει το a και το x, ενώ η δεύτερη μας δείχνει τον τρόπο με τον οποίο το *Mathematica* βλέπει το άθροισμα a+x.

```
a + x

a + x
```

```
FullForm@a + xD

Plus@a, xD
```

Κάτι ανάλογο συμβαίνει και με τις υπόλοιπες πράξεις. Π.χ. τον πολλαπλασιασμό.

```
FullForm@a  xD

Times@a, xD
```

Επομένως, θα μπορούσαμε, να πούμε ότι τα σύμβολα των πράξεων που χρησιμοποιούμε είναι ουσιαστικά συντμήσεις των αντίστοιχων συναρτήσεων. Δηλαδή, ένας σύντομος τρόπος εισαγωγής πράξεων. Αν θέλουμε να μάθουμε τα ονόματα των συναρτήσεων, τα οποία αντιστοιχούν στα σύμβολα, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **Alias**, την οποία διαθέτει το *Mathematica* για αυτό το σκοπό.

Alias["symbol"]	Δίνει το όνομα της συνάρτησης που αντιστοιχεί στο σύμβολο symbol

```
Alias@"+"D

Plus
```

```
Alias@" "D

Times
```

```
Alias@"<"D

Less
```

Ασφαλώς, μπορούμε να χρησιμοποιήσουμε αντί των συμβόλων τις συναρτήσεις που αντιστοιχούν σε αυτά. Για παράδειγμα, αντί για το σύμβολο της πρόσθεσης μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **Plus**

```
Plus@10, 20, 30D

60
```

ή αντί για το σύμβολο του πολλαπλασιασμού να χρησιμοποιήσουμε την συνάρτηση **Times** κ.τ.λ.

```
Times@20, 30D

600
```

Είναι, όμως, προφανές ότι τα σύμβολα + και * χρησιμοποιούνται για τη δική μας ευκολία. Όπως θα δούμε τέτοιες συντμήσεις υπάρχουν και για άλλες συναρτήσεις.

Η συνάρτηση **Power**[x,n] υψώνει την παράσταση x στη δύναμη n. Μια σύντμηση της συνάρτησης αυτής είναι ο χαρακτήρας "^".

```
Power@10, 3D

1000
```

```
Alias@"^"D

Power
```

```
10^3

1000
```

Ένας άλλος τρόπος για να εισάγουμε τη δύναμη κάποιου αριθμού είναι ο συνδυασμός πλήκτρων Ctrl + 6. Στην περίπτωση αυτή, η εισαγόμενη εντολή παίρνει τη συνήθη μορφή που χρησιμοποιούμε στα Μαθηματικά. Για παράδειγμα, η επόμενη εντολή πληκτρολογήθηκε με τη σειρά 10 Ctrl+6 3.

```
10^3

1000
```

Είναι φανερό ότι κάθε εργασία στο περιβάλλον του *Mathematica* αντιμετωπίζεται με τη βοήθεια συναρτήσεων. Επομένως είναι απαραίτητο ο κάθε χρήστης να γνωρίζει, όχι μόνο τα ονόματα των συναρτήσεων, αλλά και τον τρόπο με τον οποίο χρησιμοποιούνται, και το σκοπό κάθε συνάρτησης. Το γεγονός ότι τα ονόματα των συναρτήσεων είναι ολόκληρες λέξεις, οι οποίες συνήθως μας πληροφορούν για το αποτέλεσμα των συναρτήσεων είναι σημαντικό, διότι μπορούμε σχετικά εύκολα να βρούμε την κατάλληλη συνάρτηση για αυτό που θέλουμε να κάνουμε.

Ας υποθέσουμε για παράδειγμα, ότι χρειαζόμαστε μια συνάρτηση, η οποία να επιστρέφει το ακέραιο μέρος ενός αριθμού. Οι λέξεις κλειδιά είναι Integer (ακέραιος) και Part (μέρος). Φυσικά δεν γνωρίζουμε το ακριβές όνομα της συνάρτησης, όμως θεωρούμε πιθανό ότι η συνάρτηση αυτή θα περιέχει στο όνομά της είτε τη λέξη Integer είτε τη λέξη Part. Επίσης, δεν είμαστε στη θέση να γνωρίζουμε αν η λέξη Integr (ή Part) θα εμφανίζεται στην αρχή ή στο τέλος του ονόματος. Το *Mathematica* μπορεί να παρουσιάσει όλες τις συναρτήσεις που διαθέτει και περιέχουν στο όνομά τους κάποιους συγκεκριμένους κάποιους συγκεκριμένους χαρακτήρες με εντολή που αρχίζει με τον χαρακτήρα "?".

Παρόλα αυτά το πρόβλημα της εύρεσης της συνάρτησης δεν έχει λυθεί ακόμα, γιατί εφόσον δεν γνωρίζουμε σε ποιο σημείο του ονόματος θα εμφανίζεται η λέξη Integer, είναι εξίσου πιθανό πριν από τη λέξη αυτή να υπάρχουν μηδέν ή περισσότεροι χαρακτήρες. Φυσικά μηδέν ή περισσότεροι χαρακτήρες μπορεί να υπάρχουν και μετά τη λέξη Integer. Επομένως προκύπτει η ανάγκη σωστής καθοδήγησης του *Mathematica*. Η επόμενη εντολή κινείται προς αυτή την κατεύθυνση.

```
?*Integer*
```

```
FactorInteger    IntegerDigits    IntegerQ
GaussianIntegers IntegerExponent  Integers
Integer          IntegerPart
```

Ο τρόπος με τον οποίο θα εμφανιστεί η απάντηση εξαρτάται από την έκδοση του προγράμματος *Mathematica*. Σε κάθε περίπτωση το πρόγραμμα θα εμφανίσει ένα κατάλογο με ονόματα συναρτήσεων.

Ο χαρακτήρας "*", που εμφανίζεται στην προηγούμενη εντολή, αντιμετωπίζεται από το *Mathematica* με ειδικό τρόπο. Το πρόγραμμα θα αντικαταστήσει το χαρακτήρα αυτό με μηδέν ή περισσότερους χαρακτήρες. Δηλαδή η παράσταση *Integer* σημαίνει ότι θέλουμε να μάθουμε όλες τις συναρτήσεις, οι οποίες περιέχουν σε κάποιο σημείο του ονόματός τους τη λέξη Integer. Μπορούμε αν θέλουμε να χρησιμοποιήσουμε την παράσταση Integer*, που σημαίνει ότι θέλουμε όλες τις συναρτήσεις που αρχίζουν με τη λέξη Integer. Προφανώς, η παράσταση *Integer σημαίνει ότι θέλουμε τις συναρτήσεις που τελειώνουν με την λέξη Integer>

Στο κατάλογο με τις συναρτήσεις, τις οποίες παρουσίασε το *Mathematica*, βλέπουμε ότι υπάρχει και η συνάρτηση IntegerPart. Ασφαλώς, μοιάζει να είναι η συνάρτηση, την οποία αναζητούμε, όμως, αν δεν είμαστε σίγουροι ότι αυτή είναι, μπορούμε να ζητήσουμε βοήθεια από το ίδιο το *Mathematica*. Ο χαρακτήρας "?" όταν ακολουθείται από το όνομα συγκεκριμένης συνάρτησης, είναι μια οδηγία προς το *Mathematica* να παρουσιάσει βοήθεια για τη συγκεκριμένη συνάρτηση.

```
?IntegerPart
```

```
IntegerPart@xD gives the integer part of x.
```

Ο τρόπος με τον οποίο θα εμφανιστεί η απάντηση εξαρτάται, και πάλι, από την έκδοση του προγράμματος *Mathematica*. Σε κάθε περίπτωση το πρόγραμμα θα εμφανίσει ένα σύντομο κείμενο βοήθειας που επεξηγεί το σκοπό της συνάρτησης και τον τρόπο σύνταξής της.

Από το σύντομο αυτό κείμενο βοήθειας είμαστε πλέον βέβαιοι ότι πρόκειται για τη συνάρτηση, την οποία αναζητούσαμε.

```
IntegerPart@PiD

3
```

Εκτός από τη συνάρτηση IntegerPart το πρόγραμμα παρουσίασε και τη συνάρτηση FactorInteger. Αν και το όνομα δείχνει το σκοπό της συνάρτησης αυτής, εντούτοις μπορούμε να ζητήσουμε βοήθεια από το πρόγραμμα.

```
?FactorInteger
```

```
FactorInteger@nD gives a list of the prime factors
  of the integer n, together with their exponents.
```

Διαπιστώνουμε ότι η συνάρτηση FactorInteger αναλύει έναν ακέραιο σε γινόμενο πρώτων παραγόντων.

```
FactorInteger@1358280D

882, 3<, 83, 2<, 85, 1<, 87, 3<, 811, 1<<
```

1.2 Κατασκευή Συναρτήσεων

Αν και υπάρχουν εκατοντάδες ενσωματωμένες στο *Mathematica* συναρτήσεις, μερικές φορές δεν υπάρχει κάποια κατάλληλη για να μας δώσει το αποτέλεσμα που θέλουμε. Στην περίπτωση αυτή μπορούμε (α) είτε να συνδυάσουμε διάφορες συναρτήσεις του *Mathematica*, (β) είτε να κατασκευάσουμε μία δική μας.

Η κατασκευή συναρτήσεων μπορεί να γίνει με διάφορους τρόπος. Ας δούμε μερικά απλά παραδείγματα.

Παράδειγμα 1: Να κατασκευαστεί μια συνάρτηση η οποία να επιστρέφει το αποτέλεσμα της δύναμης x^x, για κάποιο ακέραιο x.

Ένας τρόπος κατασκευής αυτής της συνάρτηση είναι ο εξής:

```
r = H#^#L &

#1^#1 &
```

Το όνομα της συνάρτησης είναι r ενώ το κυρίως σώμα της συνάρτησης είναι η παράσταση (# ^ #) &. Οι παρενθέσεις χρησιμοποιούνται μόνο για λόγους ομαδοποίησης, και θα μπορούσαν να παραληφθούν χωρίς να χαθεί τίποτε από τη συνάρτηση. Το σύμβολο & δείχνει ότι αυτό που προηγείται πρέπει να αντιμετωπιστεί ως συνάρτηση. Είναι προφανές ότι είναι ένα σημαντικό σύμβολο, και δεν πρέπει να παραληφθεί. Τέλος το σύμβολο # δηλώνει τη θέση στην οποία το *Mathematica* θα τοποθετήσει το όρισμα της συνάρτησης, δηλαδή το *Mathematica* θα αντικαταστήσει το σύμβολο αυτό με το όρισμα, το οποίο θα δώσουμε στη συνάρτηση. Π.χ. αν το όρισμα της συνάρτησης είναι ο αριθμός 2, τότε το *Mathematica* θα αντικαταστήσει το σύμβολο # με τον αριθμό 2, δηλαδή το πρόγραμμα θα μας δώσει τη δύναμη 2^2.

Το σύμβολο της ισότητας (=) δεν έχει καμία σχέση με το σύμβολο της μαθηματικής ισότητας, που χρησιμοποιούμε στις εξισώσεις. Ουσιαστικά είναι μια αντικατάσταση του αριστερού μέρους της ισότητας αυτής με το δεξιό. Το *Mathematica* κάθε φορά που θα βλέπει το γράμμα r, θα το αντικαθιστά αυτόματα με το δεξίο μέρος της ισότητας αυτής.

Στη συνέχεια βλέπουμε τις τιμές που επιστρέφει η συνάρτηση r για συγκεκριμένες τιμές (ορίσματα):

```
r@2D

4
```

```
r@10D

10000000000
```

```
r@xD
```

$$x^x$$

```
r@bD
```

$$b^b$$

Παρατηρούμε ότι η εκτέλεση της συνάρτησης r μας δίνει ακριβώς αυτό που ζητάμε.

Επίσης παρατηρούμε ότι στην πρώτη εκτέλεση της συνάρτησης r το *Mathematica* έδωσε μια απάντηση της μορφής $\#1^{\#1}$ &. Το σύμβολο #1 δείχνει ότι το πρόγραμμα θεωρεί ότι υπάρχει μόνο μία θέση η οποία πρέπει να συμπληρωθεί από το όρισμα της συνάρτησης. Επειδή στον εκθέτη θέλουμε τον ίδιο αριθμό (δηλαδή το ίδιο όρισμα), το *Mathematica* τοποθέτησε και στον εκθέτη το ίδιο σύμβολο.

Παράδειγμα 2: Να κατασκευαστεί μια συνάρτηση η οποία να υπολογίζει τη δύναμη x^y.

```
s = H#1^#2L &
```

$$\#1^{\#2}\ \&$$

Επειδή στη συγκεκριμένη συνάρτηση έχουμε άλλον αριθμό στον εκθέτη χρησιμοποιήσαμε και μια δεύτερη θέση (#2). Στη συνέχεια βλέπουμε τις τιμές που επιστρέφει η συνάρτηση s για συγκεκριμένες τιμές:

```
s@2, 3D
```

8

```
s@3, 2D
```

9

```
s@x, yD
```

$$x^y$$

Παρατηρούμε ότι η εκτέλεση της συνάρτησης s μας δίνει ακριβώς αυτό που ζητάμε.

Το *Mathematica* κρατάει στη μνήμη του οτιδήποτε εκτελεί ο χρήστης, μέχρι την έξοδο του προγράμματος.

Επομένως είναι χρήσιμο, για λόγους αποσυμφόρησης της μνήμης, αλλά κυρίως για την αποφυγή λαθών, να διαγράφουμε κάποιες μεταβλητές τις οποίες δεν χρειαζόμαστε πλέον. Η συνάρτηση **Clear** είναι ακριβώς γι αυτή τη χρήση.

```
Clear@r, sD
```

Μετά την εκτέλεση αυτής της εντολής δεν μπορούμε πλέον να χρησιμοποιήσουμε τις συναρτήσεις r και s.

```
r@2D

r@2D
```

```
s@2, 3D

s@2, 3D
```

Ένας άλλος τρόπος κατασκευής συναρτήσεων είναι με αναγραφή συγκεκριμένων μεταβλητών, οι οποίες θα αντικατασταθούν κατά την εκτέλεση της συνάρτησης.

Παράδειγμα 3: Να κατασκευαστεί συνάρτηση, η οποία θα υπολογίζει το τετράγωνο ενός αριθμού x.

Χρειαζόμαστε μία μεταβλητή για να ορίσουμε τη συνάρτηση, εφόσον ο εκθέτης παραμένει σταθερός, ίσος με 2. Η συνάρτηση θα έχει την εξής μορφή:

```
r@x_D := x^2
```

Προφανώς το όνομα της συνάρτησης είναι r και δέχεται ένα όρισμα που συμβολίζεται με τον χαρακτήρα x. Με τον χαρακτήρα υπογράμμισης _ που ακολουθεί το x είναι το *Mathematica* θεωρεί το x σαν μεταβλητή την οποία θα αντικαταστήσει με το όρισμα της συνάρτησης. Δηλαδή αν ο χρήστης εκτελέσει την εντολή r[2], το *Mathematica* θα αντικαταστήσει πρώτα τη συάρτηση r με το δεξιό μέρος της ισότητας και στη συνέχεια όπου βλέπει x θα το αντικαθιστά με τον αριθμό 2.

Όπως και το σύμβολο = στον προηγούμενο τρόπο κατασκευής συνάρτησης, έτσι και το σύμβολο := είναι ένα σύμβολο αντικατάστασης του αριστερού μέρους της ισότητας με το δεξιό. Επιπλέον με το σύμβολο := το *Mathematica* διαβάζει τα δεδομένα που του δίνουμε, χωρίς να εκτελεί πράξεις που πιθανόν να υπάρχουν στην παράσταση που δίνουμε. Επειδή δεν γίνονται πράξεις το *Mathematica* δεν θα παρουσιάσει κανένα αποτέλεσμα. Έτσι εξηγείται και η έλλειψη του αποτελέσματος που αντιστοιχεί στην προηγούμενη εντολή.

Στη συνέχεια βλέπουμε τις τιμές που επιστρέφει η συνάρτηση r για συγκεκριμένες τιμές (ορίσματα):

```
r@2D

4
```

```
r@xD
```

x^2

```
r@aD
```

a^2

Παρατηρούμε ότι η εκτέλεση της συνάρτησης r μας δίνει ακριβώς αυτό που ζητάμε.

Ασφαλώς η ίδια συνάρτηση μπορεί να κατασκευαστεί με χρήση της προηγούμενου τρόπου κατασκευής συνάρτησης. Πράγματι μπορούμε να ορίσουμε τη συνάρτηση rr, σύμφωνα με τον προηγούμενο τρόπο κατασκευής συνάρτησεις, ως εξής:

```
rr = H#^2L &
```

$\#1^2$ &

Παρατηρούμε ότι η συνάρτηση rr επιστρέφει τις ίδιες τιμές που επιστρέφει και η συνάρτηση r για ίδιες συγκεκριμένες τιμές:

```
rr@2D
```

4

```
rr@xD
```

x^2

```
rr@aD
```

a^2

Είναι φανερό ότι μπορούμε να εισάγουμε δύο ή περισσότερες μεταβλητές σε μία συνάρτηση, με τόν ίδιο τρόπο που εισάγουμε μια μεταβλητή.

Παράδειγμα 4: Να κατασκευαστεί συνάρτηση, η οποία θα υπολογίζει τη n-οστή ρίζα ενός αριθμού x.

Για να κατασκευάσουμε τη συνάρτηση αυτή χρειαζόμαστε δύο μεταβλητές και η συνάρτηση θα έχει την εξής μορφή:

```
t@x_, n_D := x^H1ê nL
```

Η εκτέλεση της συνάρτησης t μας δίνει ακριβώς αυτό που ζητάμε:

```
t@25, 2D

5
```

Ηίδια συνάρτηση μπορεί να κατασκευαστεί με χρήση της προηγούμενου τρόπου κατασκευής συνάρτησης. Πράγματι μπορούμε να ορίσουμε τη συνάρτηση tt, ως εξής:

```
tt = H#1^H1ê #2LL &

#1^1ê#2 &
```

Παρατηρούμε ότι η συνάρτηση tt επιστρέφει τις ίδιες τιμές που επιστρέφει και η συνάρτηση t για ίδιες συγκεκριμένες τιμές:

```
tt@25, 2D

5
```

Κεφάλαιο 2ο: Λίστες - Πολυώνυμα

2.1 Λίστες

Οι λίστες παίζουν σημαντικό ρόλο στη χρήση συναρτήσεων του *Mathematica*. Συχνά, οι απαντήσεις που δίνει το πρόγραμμα κατά την εκτέλεση συναρτήσεων έχουν τη μορφή λίστας. Ως προς τον συμβολισμό, η λίστα μοίαζει με ένα πεπερασμένο σύνολο, δηλαδή έχει τη μορφή {1, 2, 3, a, b, 4, 5}. Όπως βλέπουμε τα μέλη της λίστα μπορεί να είναι είτε αριθμοί είτε γράμματα. Στην πραγματικότητα μέλη της λίστας μπορεί να είναι οποιαδήποτε αντικείμενα, ακόμη και συναρτήσεις του *Mathematica* ή και άλλες λίστες. Για παράδειγμα, η λίστα {1, 4, Plus[x,y], {a,b}} είναι αποδεκτή, και τα μέλη της είναι οι αριθμοί 1 και 4, η συνάρτηση Plus[x, y] και η λίστα {a, b}.

```
81, 4, Plus@x, yD, 8a, b<<

81, 4, x + y, 8a, b<<
```

Μια λίστα, ανεξάρτητα από τη μορφή με την οποία εμφανίζεται στο παράθυρο εργασίας, είναι και αυτή μία συνάρτηση. Αυτό σημαίνει ότι, ενώ ο χρήστης βλέπει τη λίστα με τον τρόπο που περιγράψαμε, το *Mathematica* αναγνωρίζει τη λίστα με διαφορετική μορφή. Πράγματι αν χρησιμοποιήσουμε τη συνάρτηση **FullForm** έχουμε:

```
FullForm@%D

List@1, 4, Plus@x, yD, List@a, bDD
```

Παρατηρούμε ότι η πλήρη μορφή μιας λίστας δεν διαφέρει πολύ από την πλήρη μορφή ενός αθροίσματος ή γινομένου. Η μόνη διαφορά που υπάρχει είναι η διαφορετική επικεφαλίδα, δηλαδή το πρόθεμα List, Plus και Times που έχει η λίστα, το άθροισμα και το γινόμενο αντίστοιχα. Αυτό σημαίνει ότι αν μπορέσουμε να αλλάξουμε την επικεφαλίδα List σε Plus, τότε αντί της λίστας θα πάρουμε άθροισμα.

Το *Mathematica* διαθέτει τη συνάρτηση **Head** με την οποία βρίσκει την επικεφαλίδα μιας παράστασης:

Head[expr]: επιστρέφει την επικεφαλίδα της παράστασης expr

```
Head@%D

List
```

```
Head@x + yD

Plus
```

Η συνάρτηση **Head** μας επιστρέφει την επικεφαλίδα, δηλαδή το πρόθεμα της παράστασης, ακόμα και για πολύπλοκες παραστάσεις:

```
FullForm@Hx + yL^zD

Power@Plus@x, yD, zD
```

```
Head@%D

Power
```

Το *Mathematica* διαθέτει τη συνάρτηση **Apply** με την οποία αντικαθιστά την επικεφαλίδα μιας παράστασης:

Apply[f, expr]: αντικαθιστά την επικεφαλίδα της παράστασης expr με την επικεφαλίδα f

Έστω μία λίστα με αριθμούς:

```
810, 20, 30<

810, 20, 30<
```

Η πλήρης μορφή της λίστας είναι:

```
FullForm@%D

List@10, 20, 30D
```

Χρησιμοποιώντας την συνάρτηση **Apply**, αλλάζουμε την επικεφαλίδα της λίστας από List σε Plus και Times:

```
Apply@Plus, %D

60
```

```
Apply@Times, %8D

6000
```

Παρατηρούμε ότι τα αποτελέσματα είναι το άθροισμα και το γινόμενο των μελών της λίστας.

Το *Mathematica* είναι εφοδιασμένο με διάφορες συναρτήσεις, με τις οποίες μπορούμε εύκολα να χειριστούμε τις λίστες. Η συνάρτηση **Part** είναι χρήσιμη όταν θέλουμε να αναφερθούμε σε κάποιο μέλος μιας λίστας. Ο συμβολισμός [[ ]] αποτελεί συντομογραφία της συνάρτησης **Part**.

Part[expr, n]:	επιστρέφει το n-οστό στη σειρά μέλος της παράστασης expr
expr[[n]]:	είναι ισοδύναμη με την παράσταση Part[expr, n]
Part[expr, -n]:	επιστρέφει το n-οστό στη σειρά μέλος της παράστασης expr μετρώντας από το τέλος
expr[[-n]]:	είναι ισοδύναμη με την παράσταση Part[expr, -n]
Part[expr, 0]:	επιστρέφει την επικεφαλίδα της παράστασης expr
expr[[0]]:	είναι ισοδύναμη με την παράσταση Part[0]
Part[expr, $\{i_1, i_2, ..., i_k\}$]:	επιστρέφει μια λίστα της οποίας τα στοιχεία είναι τα $i_1, i_2, ..., i_k$ μέλη της παράστασης expr
expr[[$\{i_1, i_2, ..., i_k\}$]]:	είναι ισοδύναμη με την παράσταση Part[expr, $\{i_1, i_2, ..., i_k\}$]

Έστω η λίστα με όνομα a:

```
a = 81, 2, 10, x, y, 84, 5, 6<, 810, 20, 30<<

81, 2, 10, x, y, 84, 5, 6<, 810, 20, 30<<
```

Παρατηρούμε ότι μέλη της λίστας είναι αριθμοί, γράμματα αλλά και άλλες λίστες.

Στη συνέχεια βλέπουμε μερικές εφαρμογές της συνάρτησης **Part** και του ισοδύναμου συμβολισμού [[ ]]:

```
Part@a, 3D

10
```

```
a@@3DD

10
```

Παρατηρούμε ότι επιστρέφεται ως αποτέλεσμα το τρίτο μέλος της λίστας που είναι ο αριθμός 3.

```
Part@a, −2D

84, 5, 6<
```

```
a@@−2DD

84, 5, 6<
```

Παρατηρούμε ότι επιστρέφεται ως αποτέλεσμα το δεύτερο μέλος της λίστας, μετρώντας από το τέλος, που είναι η λίστα {4, 5, 6}.

```
Part@a, 0D

List
```

```
a@@0DD

List
```

Παρατηρούμε ότι επιστρέφεται ως αποτέλεσμα η επικεφαλίδα της παράσταση a, η οποία είναι List αφού η παράσταση a είναι μια λίστα.

Μπορούμε ακόμη, χρησιμοποιώντας την τελευταία επιλογή της συνάρτησης **Part**, να σχηματίσουμε μια άλλη λίστα επιλέγοντας συγκεκριμένα στοιχεία της λίστας a:

```
Part@a, 81, 3, 5<D

81, 10, y<
```

```
a@@81, 3, 5<DD

81, 10, y<
```

Παρατηρούμε ότι επιστρέφεται ως αποτέλεσμα μια λίστα με το πρώτο, τρίτο και πέμπτο μέλος της λίστα a.

Επίσης μπορούμε να σχηματίσουμε μια άλλη λίστα επιλέγοντας συγκεκριμένα στοιχεία της λίστας a με την εξής εντολή:

```
8Part@a, 3D, Part@a, 5D, Part@a, 6D<
```

```
810, y, 84, 5, 6<<
```

```
8a@@3DD, a@@5DD, a@@6DD<
```

```
810, y, 84, 5, 6<<
```

Παρατηρούμε ότι επιστρέφεται ως αποτέλεσμα μια λίστα με το τρίτο, πέμπτο και έκτο μέλος της λίστα a.

Η λίστα {4, 5, 6} θεωρείται μέλος της λίστας a, μάλιστα είναι το έκτο μέλος της λίστας α, και για αυτό μπορούμε να αναφερθούμε σε αυτή. Ο αριθμός 5, όμως, δεν είναι μέλος της λίστας a, επομένως δεν μπορούμε να αναφερθούμε σε αυτόν με τον τρόπο που περιγράψαμε πιο πάνω. Παρατηροούμε, ότι ο αριθμός 5 είναι μέλος μιας επιμέρους λίστας, η οποία είναι το έκτο μέλος της λίστας a, ενώ ο αριθμός 5 είναι το δεύτερο μέλος της επιμέρους λίστας. Έτσι, χρησιμοποιώντας, δύο δείκτες μπορούμε να αναφερθούμε στον αριθμό 5. Με τις επόμενες εντολές ζητάμε από το *Mathematica* να μας παρουσιάσει από το έκτο μέλος της λίστας a, το δεύτερο μέλος.

```
Part@a, 6, 2D
```

```
5
```

```
a@@6, 2DD
```

```
5
```

2.2 Πίνακες και Λίστες

Όπως είδαμε παραπάνω, μπορούμε να σχηματίσουμε μια άλλη λίστα επιλέγοντας συγκεκριμένα στοιχεία της λίστας a:

```
Part@a, 86, 7<D
```

```
884, 5, 6<, 810, 20, 30<<
```

```
a@@86, 7<DD

884, 5, 6<, 810, 20, 30<<
```

Παρατηρούμε ότι μπορούμε να έχουμε λίστες, των οποίων τα μέλη είναι αποκλειστικά λίστες. Στην πραγματικότητα ένας mxn πίνακας ορίζεται σαν μια λίστα, η οποία αποτελείται από m επιμέρους λίστες, τις γραμμές του πίνακα, καθεμία από τις οποίες περιέχει n στοιχεία. Το *Mathematica* διαθέτει την συνάρτηση **MatrixForm** με την οποία δείχνει τον πίνακα στη συνήθη μορφή του:

MatrixForm[list]: εμφανίζει τα στοιχεία της λίστας list ως πίνακα με τη συνήθη μορφή.

```
MatrixForm@%D

J 4   5   6 N
  10  20  30
```

Ένα από τα μειονεκτήματα που έχουν οι λίστες είναι ότι για να εισαχθούν στο *Mathematica* πρέπει να αναγραφούν τα στοιχεία τους ένα προς ένα. Για το λόγο αυτό το πρόγραμμα είναι εφοδιασμένο με κάποιες συναρτήσεις, οι οποίες βοηθούν στην κατασκευή λιστών και κατ' επέκταση πινάκων. Μια από αυτές τις συναρτήσεις είναι η συνάρτη **Range**:

Range[imax]: επιστρέφει τη λίστα {1, 2, 3, ..., imax}.

Range[imin,imax]: επιστρέφει τη λίστα {imin, imin+1, imin+2, ..., imaz-1, imax}.

Range[imin,imax,step]: επιστρέφει τη λίστα {imin, imin+step, imin+2*step, ..., **§** imax}.

Είναι προφανές ότι στις δύο πρώτες περιπτώσεις η συνάρτηση **Range** δημιουγεί μια λίστα με διαδοχικούς ακεραίους,

```
Range@10D

81, 2, 3, 4, 5, 6, 7, 8, 9, 10<
```

```
Range@3, 9D

83, 4, 5, 6, 7, 8, 9<
```

ενώ στην τρίτη περίπτωση ελέγχουμε το βήμα αύξησης step των στοιχείων της λίστας. Στην τελευταία περίπτωση είναι δυνατόν το άνω όριο imax να μην υπάρχει στην λίστα, διότι δεν το επιτρέπει το βήμα που καθορίσαμε. Για παράδειγμα, στη λίστα:

```
Range@1, 20, 4D

81, 5, 9, 13, 17<
```

δεν περιέχεται το άνω όριο, αφού το άθροισμα 17 + 4 ξεπερνά το όριο αυτό. Επομένως πρέπει να είμαστε προσεκτικοί, όταν καθορίζουμε το βήμα αύξησης, και θέλουμε οπωσδήποτε το άνω όριο να περιέχεται στη λίστα που θα δημιουργηθεί.

Η συνάρτηση **Range** μπορεί να χρησιμοποιηθεί και για την κατασκευή πινάκων:

```
8Range@4D, Range@7, 10D, Range@10, 20, 3D<

881, 2, 3, 4<, 87, 8, 9, 10<, 810, 13, 16, 19<<
```

Παρατηρούμε

```
MatrixForm@%D

 1   2   3   4
 7   8   9  10
10  13  16  19
```

Μια άλλη συνάρτηση σχηματισμού λιστών και κατ' επέκταση πινάκων είναι η **Table**:

Table[expr, {imax}]: σχηματίζει μία λίστα με imax αντίγραφα της παράστασης expr.

Table[expr, {i, imax}]: σχηματίζει μία λίστα από τις τιμές της παράστασης expr, όταν το i παίρνει τιμές από 1 μέχρι imax.

Table[expr, {i, imin, imax}]: σχηματίζει μία λίστα από τις τιμές της παράστασης expr, όταν το i παίρνει τιμές από imin μέχρι imax.

Table[expr, {i, imin, imax, step}]: σχηματίζει μία λίστα από τις τιμές της παράστασης expr, όταν το i παίρνει τιμές imin, imin+step, imin+2*step, ..., **§** imax.

Table[expr, {i, imin, imax}, {j, jmin, jmax}]: σχηματίζει μία λίστα της οποίας τα στοιχεία είναι επιμέρους λίστες. Το πλήθος των επιμέρους λιστών προσδιορίζεται από το πλήθος των τιμών του δείκτη i. Ο δείκτης j καθορίζει το πλήθος των στοιχείων των επιμέρους λιστών.

Ανεξάρτητα από την πολυπλοκότητα της σύνταξης της συνάρτησης **Table**, η χρήση της είναι απλή:

```
Table@xyz, 85<D

8xyz, xyz, xyz, xyz, xyz<
```

```
Table@2^i, 8i, 8<D

82, 4, 8, 16, 32, 64, 128, 256<
```

```
Table@10 - i, 8i, 0, 10<D

810, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0<
```

```
Table@x^i, 8i, 2, 10, 2<D

8x^2, x^4, x^6, x^8, x^10<
```

Η συνάρτηση **Table** είναι ιδανική για την κατασκευή πινάκων. Η παρακάτω εντολή σχηματίζει έναν 3x4 πίνακα:

```
Table@i^j, 8i, 3<, 8j, 4<D

881, 1, 1, 1<, 82, 4, 8, 16<, 83, 9, 27, 81<<
```

Ας δούμε με ποιο τρόπο εκτελείται η παραπάνω εντολή. Η συνάρτηση **Table** αρχίζει με τον πρώτο δείκτη, δηλαδή τον δείκτη i, στον οποίο δίνει την πρώτη τιμή, δηλαδή i = 1. Αμέσως μετά πηγαίνει στον δεύτερο δείκτη, δηλαδή τον j, στον οποίο δίνει με τη σειρά όλες τις δυνατές τιμές που αντιστοιχούν στο δείκτη αυτό, δηλαδή j = 1, j = 2, j = 3 και j = 4. Αυτό σημαίνει ότι τα ζεύγη (i, j) που θα πάρουμε είναι (1, 1), (1, 2), (1, 3) και (1, 4). Τα ζεύγη αυτά τα αντικαθιστά στην παράσταση i ^ j με αποτέλεσμα τις τιμές 1^1 **=** 1, 1^2 **=** 1, 1^3 **=** 1και 1^4 **=** 1 οι οποίες αποτελούν τα στοιχεία της πρώτης επιμέρους λίστας. Στη συνέχεια, ο δείκτης i θαπάρει την επόμενη τιμή, δηλαδή i = 2, ενώ ο δείκτης j θα ξαναπάρει όλες τις δυνατές τιμές. Έτσι, σχηματίζονται τα ζεύγη (2, 1), (2, 2), (2, 3) και (2, 4). Τα ζεύγη αυτά αντιστοιχούν στις τιμές 2^1 **=** 2, 2^2 **=** 4, 2^3 **=** 8και 2^4 **=** 16 οι οποίες αποτελούν τα στοιχεία της δεύτερης επιμέρους λίστας. Η διαδικασία αυτή συνεχίζεται μέχρι να εξαντληθούν όλες οι δυνατές τιμές του πρώτου δείκτη i.

Η συνάρτηση **MatrixForm** μπορεί να μας δώσει, τώρα, τον πίνακα στη συνήθη του μορφή:

```
MatrixForm@%D

⎛ 1  1   1   1 ⎞
⎜ 2  4   8  16 ⎟
⎝ 3  9  27  81 ⎠
```

Στο επόμενο παράδειγμα φαίνεται καθαρά ο τρόπος με τον οποίο χειρίζεται τους δείκτες η συνάρτηση **Table**:

```
Table@a@i, jD, 8i, 3<, 8j, 4<D êê MatrixForm

⎛ a@1, 1D  a@1, 2D  a@1, 3D  a@1, 4D ⎞
⎜ a@2, 1D  a@2, 2D  a@2, 3D  a@2, 4D ⎟
⎝ a@3, 1D  a@3, 2D  a@3, 3D  a@3, 4D ⎠
```

Η οδηγία "// MatrixForm" στο τέλος της προηγούμενης εντολής ζητά από το μαθημάτικα να εφαρμόσει τη συνάρτηση **MatrixForm** στο αποτέλεσμα που βρεθεί. Γενικότερα, η εντολή "x // f" θα εφαρμόσει τη συνάρτηση f στο x:

```
x êê f

f@xD
```

Είναι προφανές ότι ορισμένες μορφές της συνάρτησης **Table** είναι ισοδύναμες με τη συνάρτηση **Range**.

Δημιουργία λίστας με τη συνάρτηση **Table**[expr, {i, imax}]:

```
Table@i, 8i, 10<D

81, 2, 3, 4, 5, 6, 7, 8, 9, 10<
```

Δημιουργία λίστας με τη συνάρτηση **Range**[imax]:

```
Range@10D

81, 2, 3, 4, 5, 6, 7, 8, 9, 10<
```

Έλεγχος αν οι δύο προηγούμενες εντολές είναι ισοδύναμες:

```
Table@i, 8i, 10<D « Range@10D

True
```

Ο συμβολισμός == είναι συντομογραφία της συνάρτησης **Equal**, και ταυτόχρονα αποτελεί το σύμβολο της μαθηματικής ισότητας. Η παραπάνω εντολή είναι ουσιαστικά μία ερώτηση προς το *Mathematica*, αν το αριστερό μέλος είναι ίσο με το δεξιό μέλος της ισότητας. Όπως βλέπουμε η απάντηση είναι καταφατική (True).

Ο μαθηματικός συμβολισμός ≠, ο οποίος είναι συντομογραφία της συνάρτησης **Unequal**, εισάγεται στο *Mathematica* με τον συνδυασμό !=. Όπως και στην περίπτωση της ισότητας, η εντολή x != y είναι μια ερώτηση προς το *Mathematica* αν ισχύει η μαθηματική σχέση x ≠ y.

```
Table@i, 8i, 10<D != Range@10D

False
```

Όταν συγκρίνουμε λίστες πρέπει να είμαστε προσεκτικοί, διότι το *Mathematica* δεν αντιμετωπίζει τις λίστες ως σύνολα. Για παράδειγμα, ίσως εμείς να θεωρούμε ότι οι παρακάτω λίστες, οι οποίες κατασκευάζονται με τη χρήση των συναρτήσεων **Table** και **Range**, ταυτίζονται

```
Table@10 - i, 8i, 0, 5<D

810, 9, 8, 7, 6, 5<
```

```
Range@5, 10D

85, 6, 7, 8, 9, 10<
```

όμως το *Mathematica* βλέπει τα πράγματα διαφορετικά.

```
Table@10 - i, 8i, 0, 5<D ≠ Range@5, 10D

True
```

Το παραπάνω παράδειγμα δείχνει ότι το *Mathematica* αντιμετωπίζει τις λίστες σαν διατεταγμένα σύνολα, οπότε η διάταξη των στοιχείων μιας λίστας παίζει σημαντικό ρόλο.

Οι συναρτήσεις **Range** και **Table** είναι χρήσιμες όταν θέλουμε να κατασκευάσουμε λίστες και κατ' επέκταση πίνακες, των οποίων τα στοιχεία ικανοποιούν κάποιους κανόνες. Όταν, όμως, θέλουμε λίστες με συγκεκριμένα στοιχεία, τα οποία δεν ακολουθούν κάποιο κανόνα, τότε ασφαλώς πρέπει να εισάγουμε τα

στοιχεία ένα προς ένα.

Μερικές φορές, όμως, χρειαζόμαστε μια λίστα με τυχαία στοιχεία. Δηλαδή, μία λίστα της οποίας τα μέλη δεν είναι κάποια συγκεκριμένα στοιχεία αλλά ούτε ικανοποιούν κάποια σχέση. Ο ευκολότερος τρόπος για να πάρουμε μια τέτοια λίστα είναι να χρησιμοποιήσουμε τη συνάρτηση **Random**:

Random[ ]:	επιστρέφει ένα τυχαίο πραγματικό αριθμό μεταξύ 0 και 1.
Random[type]:	επιστρέφει ένα τυχαίο αριθμό τύπου type μεταξύ 0 και 1. Ο τύπος (type) του αριθμού μπορεί να είναι: Integer, Real, Complex.
Random[type, range]:	επιστρέφει ένα τυχαίο αριθμό του συγκεκριμένου τύπου type ο οποίος βρίσκεται στο πεδίο range. Ο τύπος (type) του αριθμού μπορεί να είναι: Integer, Real, Complex. Το πεδίο (range) μπορεί να είναι της μορφής {imin, imax} ή απλά imax που είναι ισοδύναμο με το πεδίο {0, imax}.

Η συνάρτηση **Random** είναι ένα παράδειγμα, το οποίο δείχνει ότι μια συνάρτηση μπορεί να έχει από μηδέν έως περισσότερα ορίσματα. Επίσης, παρατηρούμε ότι και στην περίπτωση που δεν υπάρχει κανένα όρισμα, οι αγκύλες [ ] συνοδεύουν το όνομα της συνάρτησης.

Η εντολή που ακολουθεί είναι μία λίστα τεσσάρων εντολών. Η πρώτη θα δώσει ένα τυχαίο πραγματικό αριθμό μεταξύ 0 και 1, η δεύτερη ένα τυχαίο ακέραιο μεταξύ 0 και 1, η τρίτη ένα τυχαίο πραγματικό μεταξύ 0 και 5 και η τελευταία ένα τυχαίο ακέραιο μεταξύ 12 και 34.

```
8Random@D, Random@IntegerD,
 Random@Real, 5D, Random@Integer, 812, 34<D<

80.269114, 0, 3.44615, 27<
```

Επειδή η συνάρτηση **Random** επιστρέφει ένα τυχαίο αριθμό, είναι προφανές ότι κάθε φορά που εκτελούμε την προηγούμενη εντολή, θα παίρνουμε μια διαφορετική λίστα τεσσάρων αριθμών.

Αν η συνάρτηση **Random** συνδυαστεί με τη συνάρτηση **Table**, μπορεί να μας δώσει ένα τυχαίο πίνακα με το μέγεθος που θέλουμε. Ας υποθέσουμε, για παράδειγμα, ότι θέλουμε ένα πίνακα 3x3 με στοιχεία ακεραίους μεταξύ 5 και 15:

```
b = Table@Random@Integer, 85, 15<D, 8i, 1, 3<, 8j, 1, 3<D

889, 13, 9<, 813, 9, 5<, 88, 11, 7<<
```

```
MatrixForm@%D

 7  8  5
15  8  6
 5  7  5
```

Όπως αναφέραμε και προηγουμένως, η συνάρτηση **Random** επιστρέφει ένα διαφορετικό τυχαίο αριθμό κάθε φορά που εκτελείται. Αυτό σημαίνει ότι αν εκτελέσουμε την ίδια εντολή που καθόρισε τον πίνακα b, τότε ο πίνακας αυτός θα αλλάξει:

```
c = Table@Random@Integer, 85, 15<D, 8i, 1, 3<, 8j, 1, 3<D

8813, 13, 10<, 88, 10, 15<, 87, 13, 8<<
```

```
MatrixForm@%D

13  13  10
 8  10  15
 7  13   8
```

Έτσι, έχουμε ουσιαστικά μια μηχανή κατασκευής τυχαίων πινάκων, οι οποίοι βέβαια έχουν κάποια κοινά χαρακτηριστικά. Τα στοιχεία τους είναι ακέραιοι αριθμοί, και μάλιστα μεταξύ 5 και 15.

Τώρα, πλέον το *Mathematica* γνωρίζει δύο 3x3 πίνακες, και συγκεκριμένα τους πίνακες b και c.

Το *Mathematica* διαθέτει αρκετές συναρτήσεις με τις οποίες διαχειρίζεται τους πίνακες. Μερικές από τις οποίες είναι:

Dot[A, B] ή A.B:	επιστρέφει το γινόμενο των πινάκων A και B.
Det[A]:	επιστρέφει την ορίζουσα ενός τετραγωνικού πίνακα A.
Inverse[A]:	επιστρέφει τον αντίστροφο πίνακα ενός αντιστρέψιμου πίνακα A.
Transpose[A]:	επιστρέφει τον ανάστροφο του πίνακα A.

Χρησιμοποιώντας τις συναρτήσεις αυτές, μπορεί κάποιος εύκολα να διαχειριστεί του πίνακες b και c. Συγκεκριμένα μπορούμε να τους πολλαπλασιάσουμε:

```
b.c

88190, 236, 230<, 8301, 353, 318<, 8156, 200, 195<<
```

```
MatrixForm@%D
```

$$\begin{pmatrix} 190 & 236 & 230 \\ 301 & 353 & 318 \\ 156 & 200 & 195 \end{pmatrix}$$

Ο πολλαπλασιασμός έγινε με την τοποθέτηση μιας τελείας μεταξύ των πινάκων. Η τελεία αυτή είναι ουσιαστικά συντομογραφία της συνάρτησης **Dot**, γεγονός που εύκολα μπορεί να επαληθευθεί:

```
Alias@"."D

Dot
```

```
Dot@b, cD

88190, 236, 230<, 8301, 353, 318<, 8156, 200, 195<<
```

```
MatrixForm@%D
```

$$\begin{pmatrix} 190 & 236 & 230 \\ 301 & 353 & 318 \\ 156 & 200 & 195 \end{pmatrix}$$

Να βρούμε την ορίζουσα τους:

```
Det@bD

−49
```

```
Det@cD

−622
```

Να βρούμε τον αντίστροφό τους, εφόσον διαπιστώσουμε ότι υπάρχει:

```
Inverse@bD
```

$$\left\{\left\{\frac{2}{49}, \frac{5}{49}, -\frac{8}{49}\right\}, \left\{\frac{45}{49}, -\frac{10}{49}, -\frac{33}{49}\right\}, \left\{-\frac{65}{49}, \frac{9}{49}, \frac{64}{49}\right\}\right\}$$

```
MatrixForm@%D
```

$$\begin{pmatrix} \frac{2}{49} & \frac{5}{49} & -\frac{8}{49} \\ \frac{45}{49} & -\frac{10}{49} & -\frac{33}{49} \\ -\frac{65}{49} & \frac{9}{49} & \frac{64}{49} \end{pmatrix}$$

```
b.%
```

```
881, 0, 0<, 80, 1, 0<, 80, 0, 1<<
```

```
MatrixForm@%D
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
Inverse@cD
```

$$\left\{\left\{\frac{115}{622}, -\frac{13}{311}, -\frac{95}{622}\right\}, \left\{-\frac{41}{622}, -\frac{17}{311}, \frac{115}{622}\right\}, \left\{-\frac{17}{311}, \frac{39}{311}, -\frac{13}{311}\right\}\right\}$$

```
MatrixForm@%D
```

$$\begin{pmatrix} \frac{115}{622} & -\frac{13}{311} & -\frac{95}{622} \\ -\frac{41}{622} & -\frac{17}{311} & \frac{115}{622} \\ -\frac{17}{311} & \frac{39}{311} & -\frac{13}{311} \end{pmatrix}$$

```
c.%
```

```
881, 0, 0<, 80, 1, 0<, 80, 0, 1<<
```

```
MatrixForm@%D
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Τέλος μπορούμε να βρούμε τον ανάστροφό τους:

```
Transpose@bD

887, 15, 5<, 88, 8, 7<, 85, 6, 5<<
```

```
MatrixForm@%D

 7  15  5
 8   8  7
 5   6  5
```

```
Transpose@cD

8813, 8, 7<, 813, 10, 13<, 810, 15, 8<<
```

```
MatrixForm@%D

 13   8   7
 13  10  13
 10  15   8
```

Επίσης, μπορούμε στη διαχείριση των πινάκων να χρησιμοποιήσουμε και ήδη γνωστές συναρτήσεις του *Mathematica*, όπως τη συνάρτηση **Plus** ή την αντίστοιχη συντομογραφία της "+":

```
Plus@b, cD

8820, 21, 15<, 823, 18, 21<, 812, 20, 13<<
```

```
MatrixForm@%D

 20  21  15
 23  18  21
 12  20  13
```

```
b + c

8820, 21, 15<, 823, 18, 21<, 812, 20, 13<<
```

```
MatrixForm@%D
```

$$\begin{pmatrix} 20 & 21 & 15 \\ 23 & 18 & 21 \\ 12 & 20 & 13 \end{pmatrix}$$

Παρατήρηση: Το *Mathematica* δεν κάνει διάκριση μεταξύ διανυσμάτων γραμμών και διανυσμάτων στηλών. Η συνάρτηση **Dot** είναι κατασκευασμένη έτσι ώστε να δίνει κάθε δυνατό αποτέλεσμα.

Πράγματι, έστω:

```
p = 81, 2, 3<

81, 2, 3<
```

Η λίστα p που ορίστηκε μπορεί να θεωρηθεί και ως πίνακας αφού το *Mathematica* δεν κάνει διάκριση ανάμεσα στις λίστες και στους πίνακες. Επίσης, μπορεί να θεωρηθεί και ως διάνυσμα με συντεταγμένες 1, 2 και 3. Όμως, ο πίνακας p ή το διάνυσμα p μπορεί να θεωρηθεί είτε σαν πίνακας (διάνυσμα) γραμμή είτε σαν πίνακας (διάνυσμα) στήλη. Αν ο p θεωρηθεί σαν πίνακας (διάνυσμα) γραμμή, τότε το γινόμενο b·p δεν ορίζεται, ενώ αν θεωρηθεί σαν πίνακας (διάνυσμα) στήλη, τότε το γινόμενο p·b δεν μπορεί να οριστεί. Επομένως, σε κάθε περίπτωση ένα από τα γινόμενα b·p ή p·b δεν μπορεί ναοριστεί. Το *Mathematica*, όμως, θα δώσει απάντηση και στις δύο περιπτώσεις.

```
8p.b, b.p<

8859, 64, 40<, 862, 46, 51<<
```

Αυτό οφείλεται στο γεγονός ότι το *Mathematica* δεν κάνει διάκριση μεταξύ διανυσμάτων γραμμών και διανυσμάτων στηλών. Η συνάρτηση **Dot** είναι κατασκευασμένη έτσι ώστε να δίνει κάθε δυνατό αποτέλεσμα. Όταν, λοιπόν, πολλαπλασιάζουμε δύο διανύσματα {a, b, c} και {x, y, z} χρησιμοποιώντας την συνάρτηση **Dot**, το *Mathematica* αντιμετωπίζει το πρώτο σαν διάνυσμα γραμμή και το δεύτερο σαν διάνυσμα στήλη. Έτσι, ουσιαστικά επιστρέφει το εσωτερικό γινόμενο των δύο διανυσμάτων:

```
82, 4, 6<.8x, y, z<

2 x + 4 y + 6 z
```

Στο γινόμενο p·b που είδαμε παραπάνω, η συνάρτηση **Dot** αντιμετώπισε το διάνυσμα p ως γραμμή, οπότε το αποτέλεσμα ήταν ένας 1x3 πίνακας, ενώ στο γινόμενο b·p το διάνυσμα p αντιμετωπίστηκε ως στήλη, οπότε το αποτέλεσμα ήταν ένας πίνακας 3x1. Έτσι και στις δύο περιπτώσεις το *Mathematica* θα επιστέψει μια λίστα με τρία μέλη.

Η αντιμετώπιση αυτή είναι ικανοποιητική στις περισσότερες περιπτώσεις. Υπάρχουν, όμως, περιπτώσεις

που μπορούμε να οδηγηθούμε σε λάθος. Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να πολλαπλασιάσουμε ένα 3x1 πίνακα X=$\{x_1,x_2,x_3\}$ με ένα 1x3 πίνακα Y=$\{y_1,y_2,y_3\}$. Ασφαλώς το αποτέλεσμα θα πρέπει να είναι ένας 3x3 πίνακας.

```
X = 8x1, x2, x3<

8x1, x2, x3<
```

```
Y = 8y1, y2, y3<

8y1, y2, y3<
```

Το *Mathematica* όμως, όπως είδαμε παραπάνω, θα αντιμετωπίσει το X σαν διάνυσμα γραμμή και το Y σαν διάνυσμα στήλη, οπότε το αποτέλεσμα δεν θα είναι ένας πίνακας 3x3 αλλά το εσωτερικό γινόμενο το διανυσμάτων X και Y:

```
X.Y

x1 y1 + x2 y2 + x3 y3
```

Μπορούμε, όμως, να αναγκάσουμε το πρόγραμμα να αντιμετωπίσει τους πίνακες X και Y όπως έμεις θέλουμε, δίνοντας τους πίνακες X και Y με μορφή πολλαπλής λίστας και όχι απλής. Συγκεκριμένα:

Δίνουμε τον πίνακα X με μορφή πολλαπλής λίστας ως εξής:

```
X = 88x1<, 8x2<, 8x3<<

88x1<, 8x2<, 8x3<<
```

Παρατηρούμε, ότι έχουμε τρεις επιμέρους λίστες οι οποίες αντιστοιχούν στις τρεις γραμμές ενός πίνακα. Ο πίνακας αυτός θα έχει ως πρώτη γραμμή την πρώτη επιμέρους λίστα, δηλαδή το στοιχείο x_1, ως δεύτερη γραμμή τη δεύτερη επιμέρους λίστα, δηλαδή το στοιχείο x_2 και ως τρίτη γραμμή την τρίτη επιμέρους λίστα, δηλαδή το στοιχείο x_3. Δηλαδή θα έχουμε έναν πίνακα 3x1, ακριβώς όπως θέλαμε.

Πράγματι, ο πίνακας X στην συνήθη του μορφή είναι ένας 3x1 πίνακας:

```
MatrixForm@%D

i x1 y
j x2 z
k x3 {
```

Δίνουμε τον πίνακα X με μορφή πολλαπλής λίστας ως εξής:

```
Y = 88y1, y2, y3<<

88y1, y2, y3<<
```

Παρατηρούμε, ότι έχουμε μία επιμέρους λίστα η οποία αντιστοιχεί σε μια γραμμή ενός πίνακα. Ο πίνακας αυτός θα έχει ως πρώτη και μοναδική γραμμή την επιμέρους λίστα, δηλαδή τα στοιχεία $8x_1, x_2, x_3<$. Επομένως θα έχουμε έναν πίνακα 1x3, ακριβώς όπως θέλαμε.

Πράγματι, ο πίνακας Y στην συνήθη του μορφή είναι ένας 1x3 πίνακας:

```
MatrixForm@%D

H y1  y2  y3 L
```

Στην περίπτωση που δώσουμε τους πίνακες X και Y με την παραπάνω μορφή τους, το Mathemtica επιστρέφει ως αποτέλεσμα του γινομένου των πινάκων X και Y έναν 3x3 πίνακα:

```
X.Y

88x1 y1, x1 y2, x1 y3<, 8x2 y1, x2 y2, x2 y3<, 8x3 y1, x3 y2, x3 y3<<
```

```
MatrixForm@%D

i x1 y1  x1 y2  x1 y3 y
j x2 y1  x2 y2  x2 y3 z
k x3 y1  x3 y2  x3 y3 {
```

2.3 Συναρτήσεις για τη Διαχείριση Λιστών

Επειδή οι λίστες παίζουν σημαντικό ρόλο στο *Mathematica*, το πρόγραμμα διαθέτει αρκετές συναρτήσεις με τις οποίες κανείς μπορεί να διαχειριστεί τις λίστες.

Έστω η λίστα με όνομα a:

```
a = 81, 2, 3, 83, 4<, 85, 6, 7<, 8, 9, 10<

81, 2, 3, 83, 4<, 85, 6, 7<, 8, 9, 10<
```

Στη συνέχεια δίνονται κάποιες συναρτήσεις με τις οποίες μπορούμε να πάρουμε έναν ή περισσότερους όρους ή ακόμη και συγκεκριμένα μέρης της λίστας a:

First[expr]: επιστρέφει το πρώτο στοιχείο της παράστασης expr.

```
First@aD

1
```

Last[expr]: επιστρέφει το τελευταίο στοιχείο της παράστασης expr.

```
Last@aD

10
```

Extract[expr, list]: επιστρέφει τα στοιχεία της expr, τα οποία βρίσκονται στις θέσεις που καθορίζονται στη list.

```
Extract@a, 4D

83, 4<
```

```
Extract@a, 84, 1<D

3
```

```
Extract@a, 884, 2<, 85, 2<<D

84, 6<
```

Take[list, n]: επιστρέφει μία λίστα με τα πρώτα n στοιχεία της λίστας list.

```
Take@a, 5D

81, 2, 3, 83, 4<, 85, 6, 7<<
```

Take[list, -n]: επιστρέφει μία λίστα με τα τελευταία n στοιχεία της λίστας list.

```
Take@a, −5D

883, 4<, 85, 6, 7<, 8, 9, 10<
```

Take[list, {m, n}]: επιστρέφει μία λίστα με τα στοιχεία της λίστας list, που βρίσκονται από τη θέση m έως τη θέση n.

```
Take@a, 84, 5<D

883, 4<, 85, 6, 7<<
```

Drop[list, n]: διαγράφει τα πρώτα n στοιχεία της λίστας list.

```
b = Drop@a, 3D

883, 4<, 85, 6, 7<, 8, 9, 10<
```

Drop[list, -n]: διαγράφει τα τελευταία n στοιχεία της λίστας list.

```
Drop@b, −2D

883, 4<, 85, 6, 7<, 8<
```

Drop[list, {m, n}]: διαγράφει τα στοιχεία της λίστας list, που βρίσκονται από τη θέση m έως τη θέση n.

```
Drop@a, 84, 5<D

81, 2, 3, 8, 9, 10<
```

Drop[list, {n}]: διαγράφει το στοιχείο της λίστας list που βρίσκεται στη θέση n.

```
Drop@a, 88<D

81, 2, 3, 83, 4<, 85, 6, 7<, 8, 9<
```

Rest[expr]: επιστρέφει τη παράσταση expr, διαγράφοντας το πρώτο στοιχείο.

```
Rest@aD

82, 3, 83, 4<, 85, 6, 7<, 8, 9, 10<
```

```
Rest@bD

885, 6, 7<, 8, 9, 10<
```

2.4 Λίστες και Κατασκευή Συναρτήσεων.

Όπως είπαμε και στην αρχή του κεφαλαίου οι λίστες παίζουν σημαντικό ρόλο στη χρήση συναρτήσεων του *Mathematica*, αφού συχνά, οι απαντήσεις που δίνει το πρόγραμμα κατά την εκτέλεση συναρτήσεων έχουν τη μορφή λίστας. Επομένως οι λίστες παίζουν σημαντικό ρόλο και στη κατασκευή συνάρτησης με συνδυασμό υπαρχώντων συναρτήσεων του *Mathematica*.

Για παράδειγμα, η παρακάτω συνάρτηση primeDivisors, η οποία βρίσκει τους πρώτους διαιρέτες ενός ακεραίου, κατασκευάστηκε με συνδυασμό των συναρτήσεων **FactorInteger**, **Transpose** και **First** του *Mathematica*, δηλαδή συναρτήσεων που δίνουν αποτέλεσμα με μορφή λίστας ή διαχειρίζονται λίστες.

```
primeDivisors@x_IntegerD := First@Transpose@FactorInteger@xDDD
```

Παρατηρούμε ότι το όνομα της συνάρτησης είναι primeDivisors και ότι ακολουθείται η περιγραφική λογική του *Mathematica* με τη διαφορά ότι το πρώτο γράμμα της πρώτης λέξης δεν είναι κεφαλαίο. Με τον τρόπο αυτό διακρίνονται εύκολα οι συναρτήσεις που κατασκευάζει ο χρήστης, από αυτές που είναι ενσωματωμένες στο *Mathematica*.

Είδαμε στο προηγούμενο κεφάλαιο, ότι ο συμβολισμός x_ σημαίνει ότι το x πρέπει να αντιμετωπιστεί σαν μεταβλητή, η οποία θα αντικατασταθεί, κατά την εκτέλεση της συνάρτησης, από το όρισμα που θα δώσει ο χρήστης. Ο συμβολισμός x_Integer σημαίνει το ίδιο με την επιπλέον συνθήκη ότι το όρισμα, που θα δώσει ο χρήστης, πρέπει να έχει επικεφαλίδα Integer. Δηλαδή, γίνεται έλεγχος στο όρισμα που δίνει ο χρήστης, και η συνάρτηση εκτελείται μόνο αν το όρισμα είναι ακέραιος.

Για παράδειγμα, αν το όρισμα της συνάρτησης είναι ο ακέραιος 808500, η συνάρτηση εκτελείται και μας δίνει ως αποτέλεσμα (με μορφή λίστας) τους πρώτους διαιρέτες του ακεραίου 808500,

```
primeDivisors@808500D

82, 3, 5, 7, 11<
```

ενώ αν το όρισμα της συνάρτησης είναι ο πραγματικός αριθμός 808500.0, η συνάρτηση δεν εκτελείται

```
primeDivisors@808500.0D

primeDivisors@808500.D
```

αφού ο αριθμός 808500.0 έχει επικεφαλίδα Real.

```
Head@808500.0D

Real
```

Ας δούμε, τώρα, πως δουλεύει η συνάρτηση primeDivisors που κατασκευάσαμε:

Είναι γνωστό, από το πρώτο κεφάλαιο, ότι η συνάρτηση **FactorInteger** επιστρέφει μια λίστα λιστών της μορφής:

$$\{\{p_1, n_1\},\{p_2, n_2\},...,\{p_m, n_m\}\}$$

όπου p_i, i=1, ..., m, είναι οι πρώτοι παράγοντες του ακεραίου (που δίνεται ως όρισμα) και n_i οι αντίστοιχοι εκθέτες:

```
FactorInteger@808500D

882, 2<, 83, 1<, 85, 3<, 87, 2<, 811, 1<<
```

Επειδή η κάθε επιμέρους λίστα περιέχει δύο στοιχεία, η απάντηση που δίνει η συνάρτηση **FactorInteger** είναι ένας πίνακας με δύο στήλες. Η πρώτη στήλη περιέχει τους πρώτους παράγοντες και η δεύτερη στήλη τους αντίστοιχους εκθέτες. Επομένως αν πάρουμε τον ανάστροφο πίνακά του, με τη χρήση της συνάρτησης **Transpose**, θα βρούμε έναν πίνακα με δύο γραμμές. Η πρώτη γραμμή θα αποτελείται από τους πρώτους παράγοντες του ακεραίου και η δεύτερη γραμμή από τους αντίστοιχους εκθέτες:

```
Transpose@%D

882, 3, 5, 7, 11<, 82, 1, 3, 2, 1<<
```

Παρατηρούμε ότι ο πίνακας δίνεται με τη μορφή λίστας που περιέχει δύο επιμέρους λίστες. Η πρώτη επιμέρους λίστα αντιστοιχεί στη πρώτη γραμμή του πίνακα και η δεύτερη επιμέρους λίστα στη δεύτερη γραμμή του πίνακα. Επομένως αν στη λίστα που προέκυψε εφαρμόσουμε την συνάρτηση **First** θα έχουμε αποτέλεσμα μία λίστα με τους πρώτους διαιρέτες του ακεραίου που έχουμε δώσει ως όρισμα:

```
First@%D

82, 3, 5, 7, 11<
```

Η συνάρτηση primeDivisors κατασκευάστηκε για να δίνει τους πρώτους διαιρέτες ενός ακεραίου. Δεν είναι όμως ο μοναδικός τρόπος με τον οποίο μπορούμε να πάρουμε το αποτέλεσμα αυτό. Ας δούμε και μία δεύτερη εκδοχή της ίδιας συνάρτησης.

Το *Mathematica* διαθέτει τη συνάρτηση **Divisors** η οποία μας επιστρέφει όλους τους θετικού διαιρέτες ενός ακεραίου:

Divisors[n]: επιστρέφει μια λίστα με όλους τους θετικούς ακεραιούς οι οποίοι διαιρούν τον ακέραιο n.

```
Divisors@808500D

81, 2, 3, 4, 5, 6, 7, 10, 11, 12, 14, 15, 20, 21, 22, 25, 28, 30,
 33, 35, 42, 44, 49, 50, 55, 60, 66, 70, 75, 77, 84, 98, 100,
 105, 110, 125, 132, 140, 147, 150, 154, 165, 175, 196, 210,
 220, 231, 245, 250, 275, 294, 300, 308, 330, 350, 375, 385,
 420, 462, 490, 500, 525, 539, 550, 588, 660, 700, 735, 750,
 770, 825, 875, 924, 980, 1050, 1078, 1100, 1155, 1225, 1375,
 1470, 1500, 1540, 1617, 1650, 1750, 1925, 2100, 2156, 2310,
 2450, 2625, 2695, 2750, 2940, 3234, 3300, 3500, 3675, 3850,
 4125, 4620, 4900, 5250, 5390, 5500, 5775, 6125, 6468, 7350,
 7700, 8085, 8250, 9625, 10500, 10780, 11550, 12250, 13475,
 14700, 16170, 16500, 18375, 19250, 23100, 24500, 26950, 28875,
 32340, 36750, 38500, 40425, 53900, 57750, 67375, 73500, 80850,
 115500, 134750, 161700, 202125, 269500, 404250, 808500<
```

Ασφαλώς ανάμεσα σε αυτούς υπάρχουν και οι πρώτοι διαιρέτες του ακεραίου. Επομένως, αν μπορούσαμε να καθοδηγήσουμε το πρόγραμμα να επιλέξει από την προηγούμενη λίστα των διαιρετών του ακεραίου μόνο τους πρώτους αριθμούς, θα είχαμε το ίδιο αποτέλεσμα με τη συνάρτηση primeDivisors που έχουμε κατασκευάσει.

Το *Mathematica* διαθέτει τις συναρτήσεις **Select** και **PrimeQ**:

Select[list, crit]: επιλέγει εκείνα τα στοιχεία της λίστας list, για τα οποία η συνθήκη crit είναι αληθής.
PrimeQ[expr]: επιστρέφει True αν η expr είναι πρώτος αριθμός και False σε κάθε άλλη περίπτωση.

Ο συνδυασμός των συναρτήσεων **Select** και **PrimeQ** μπορεί να βοήθησει προς την κατεύθυνση αυτή. Συγκεκριμένα, η συνάρτηση PrimeQ μπορεί να αποτελέσει τη συνθήκη με το οποίο η συνάρτηση Select θα επιλέξει τα στοιχεία από μια λίστα:

```
Select@%, PrimeQD

82, 3, 5, 7, 11<
```

Επομένως, η συνάρτηση primeDivisors μπορεί να κατασκευαστεί και με τον παρακάτω τρόπο:

```
primeDiv@x_IntegerD := Select@Divisors@xD, PrimeQD
```

```
primeDiv@808500D

82, 3, 5, 7, 11<
```

Όπως βλέπουμε οι συναρτήσεις primeDiv και primeDivisors έχουν ακριβώς το ίδιο αποτέλεσμα. Ωστόσο η μία από τις δύο είναι γρηγορότερη από τη άλλη. Παρακολουθώντας το παρακάτω παράδειγμα, με τη συνάρτηση **Timing** να μετρά το χρόνο εκτέλεσης της κάθε συνάρτησης

```
8Timing@primeDiv@192139662355662352351765136123DD,
 Timing@primeDivisors@192139662355662352351765136123DD<

880.406 Second, 837, 383, 401, 49297, 8136893, 84293300053<<,
 80.391 Second, 837, 383, 401, 49297, 8136893, 84293300053<<<
```

παρατηρούμε ότι η συνάρτηση primeDiv χρειάζεται περισσότερο χρόνο. Αυτό είναι φυσιολογικό, διότι η συνάρτηση primeDiv χρησιμοποιεί τη συνάρτηση **Divisors** η οποία μας δίνει όλους τους διαιρέτες του ακεραίου, οι οποίοι στη συνέχεια ελέγχονται ένας προς ένας από τη συνάρτηση **PrimeQ**, για να μας επιστρέψει η συνάρτηση **Select** μόνο αυτούς που είναι πρώτοι αριθμοί. Είναι προφανές ότι δεν χρειαζόμαστε όλους τους διαιρέτες, αφού πολλοί από αυτούς δεν θα είναι πρώτοι. Αντίθετα, η συνάρτηση primeDivisors χρησιμοποιεί την συνάρτηση **FactorInteger**, η οποία μας δίνει μόνον τους πρώτους διαιρέτες του ακεραίου, οπότε δεν χρειάζεται να γίνει οποιαδήποτε επιλογή.

Το παράδειγμα αυτό δείχνει ότι έχει σημασία ο τρόπος με τον οποίο κατασκευάζεται μια συνάρτηση, διότι βλέπουμε ότι η κατασκευή προσδιορίζει και το χρόνο εκτέλεσης της συνάρτησης.

Ας δούμε ένα ακόμα παράδειγμα κατασκευής συνάρτησης.

Το *Mathematica* διαθέτει τη συνάρτηση **GCD** η οποία βρίσκει τον μέγιστο κοινό διαιρέτη ακεραίων:

GCD$[n_1, n_2, ..., n_k]$: επιστρέφει το μέγιστο κοινό διαιρέτη των ακεραίων n_i.

```
GCD@124, 568, 10^3D

4
```

Ασφαλώς δεν είναι ανάγκη να βρούμε κάποια άλλη συνάρτηση υπολογισμού του μέγιστου κοινού διαιρέτη ακεραίων. Μπορούμε, όμως, χρησιμοποιώντας γνωστές συναρτήσεις του *Mathematica* να επαληθεύσουμε τη συνάρτηση **GCD**.

Όπως είναι γνωστό, ο Μ.Κ.Δ. των ακεραίων n_1, n_2, ..., n_k είναι ο μεγαλύτερος από τους κοινούς διαιρέτες των αριθμών αυτών.

Η συνάρτηση **Divisors** μπορεί να μας δώσει τους διαιρέτες των ακεραίων που μας ενδιαφέρουν. Π.χ.

```
lst1 = Divisors@124D

81, 2, 4, 31, 62, 124<
```

```
lst2 = Divisors@568D

81, 2, 4, 8, 71, 142, 284, 568<
```

```
lst3 = Divisors@10^3D

81, 2, 4, 5, 8, 10, 20, 25, 40, 50, 100, 125, 200, 250, 500, 1000<
```

Το *Mathematica* διαθέτει τις συναρτήσεις **Intersection** και **Union** με τις οποίες βρίσκει την τομή και την ένωση λιστών:

Intersection[$list_1$, $list_2$, ..., $list_k$]: επιστρέφει μια ταξινομημένη λίστα των κοινών στοιχείων των λιστών $list_i$.

Union[$list_1$, $list_2$, ..., $list_k$]: επιστρέφει μια ταξινομημένη λίστα όλων των στοιχείων που περιέχονται σε μία τουλάχιστον από τις λίστες $list_i$.

Εφαρμόζοντας τη συνάρτηση **Intersection** στις λίστες lst1 (οι διαιρέτες του 124), lst2 (οι διαιρέτες του 568) και lst3 (οι διαιρέτες του 10^3) έχουμε ως αποτέλεσμα τη λίστα με τους κοινούς διαιρέτες των αριθμών 124, 568 και 10^3.

```
Intersection@lst1, lst2, lst3D

81, 2, 4<
```

Παίρνοντας το τελευταίο στοιχείο της προηγούμενης λίστας με τη συνάρτηση **Last**, βρίσκουμε τον Μ.Κ.Δ. των αριθμών 124, 568 και 10^3.

```
Last@%D

4
```

Αν συνδυάσουμε όλες αυτές τις συναρτήσεις, μπορούμε να έχουμε την επαλήθευση που ζητούμε:

```
GCD@124, 568, 10^3D « Last@
  Intersection@Divisors@124D, Divisors@568D, Divisors@10^3DDD

True
```

2.4 Πολυώνυμα

Είναι γνωστό ότι ένα πολυώνυμο p(x) μιας μεταβλητής x, με πραγματικούς συντελεστές, είναι μια παράσταση της μορφής:

$$p(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + ... + \alpha_n x^n$$

όπου οι συντελεστές α_0, α_1, α_2, ..., α_n είναι πραγματικοί αριθμοί. Είναι προφανές ότι θα μπορούσαμε να έχουμε πολυώνυμα με ακεραίους ή ρητούς συντελεστές.

Ο βαθμός του πολυωνύμου δεν μπορεί να προσδιορίσει το πολυώνυμο, διότι υπάρχουν πολλά πολυώνυμα με το ίδιο βαθμό. Εκείνο που προσδιορίζει πλήρως ένα πολυώνυμο είναι η ακολουθία των συντελεστών του, δηλαδή η ακολουθία α_0, α_1, α_2, ..., α_n. Πράγματι, αν υποθέσουμε ότι έχουμε την ακολουθία 2, -2, 1, 2, -3, 4, τότε το πολυώνυμο που αντιστοιχεί σε αυτή είναι:

$$2 - 2 x + x^2 + 2 x^3 - 3 x^4 + 4 x^5$$

με την προϋπόθεση ότι θα γράφουμε το πολυώνυμο αρχίζοντας πάντοτε με το σταθερό του όρο. Ενώ για το πολυώνυμο:

$$2 x^2 + 2 x^4 - 3 x^5$$

η ακολουθία των συντελεστών που αντιστοιχεί σε αυτό είναι: 0, 0, 2, 0, 2, -3.

Έστω το πολυώνυμο p1:

```
p1 = 6 + 5 x - 9 x^2 - 3 x^3 + 10 x^4 - 4 x^6

6 + 5 x - 9 x^2 - 3 x^3 + 10 x^4 - 4 x^6
```

Το *Mathematica* διαθέτει αρκετές συναρτήσεις με τις οποίες μπορούμε να διαχειριστούμε πολυώνυμα. Μερικές από αυτές είναι:

Coefficient[expr, form]: επιστρέφει τον συντελεστή του όρου form στην παράσταση expr.

Coefficient[expr, form,n]: επιστρέφει τον συντελεστή του όρου form^n στην παράσταση expr.

```
Coefficient@p1, xD

5
```

```
Coefficient@p1, x^4D

10
```

```
Coefficient@p1, x, 4D

10
```

```
Coefficient@p1, x^2, 2D

10
```

Variables[poly]: επιστρέφει μια λίστα με όλες τις μεταβλητές του πολυωνύμου poly.

```
Variables@p1D

8x<
```

```
p2 = x + 2 y^2 + x^2 y^3

x + 2 y^2 + x^2 y^3
```

```
Variables@p2D

8x, y<
```

CoefficientList[poly, var]: επιστρέφει μια λίστα με τους συντελεστές των δυνάμεων της μεταβλητής var στο πουλώνυμο poly, αρχίζοντας από το σταθερό όρο.

```
CoefficientList@p1, xD

86, 5, −9, −3, 10, 0, −4<
```

```
CoefficientList@p2, xD

82 y^2, 1, y^3<
```

```
CoefficientList@p2, yD

8x, 0, 2, x^2<
```

Όταν θέλουμε να εισάγουμε ένα συγκεκριμένο πολυώνυμο στο *Mathematica* πρέπει να πληκτρολογήσουμε ένα προς ένα όλους τους όρους του. Υπάρχουν, όμως, περιπτώσεις που θέλουμε ένα τυχαίο πολυώνυμο, π,χ, με ακέραιους συντελεστές και κάποιου βαθμού. Στις περιπτώσεις αυτές θα μπορούσαμε να αποφύγουμε τη πληκτρολόγηση όλων των όρων, αν είχαμε μία "μηχανή" παραγωγής πολυωνύμων. Μια τέτοια μηχανή μπορεί να κατασκευαστεί συνδυάζοντας γνωστές συναρτήσεις του *Mathematica*.

Έστω ότι θέλουμε ένα τυχαίο πολυώνυμο με ακέραιους συντελεστές μεταξύ -5 και 5 και 6ου βαθμού. Μια πρώτη εκτίμηση για ένα τέτοιο πολυώνυμο θα είναι:

```
Table@Random@Integer, 8−5, 5<D x^k, 8k, 0, 6<D

83, −x, −5 x^2, −5 x^3, −4 x^4, 4 x^5, −2 x^6<
```

Παρατηρούμε ότι το αποτέλεσμα που μας επέστρεψε το *Mathematica* δεν είναι ένα πολυώνυμο, αλλά μια λίστα, της οποίας τα μέλη είναι οι όροι κάποιου πολυωνύμου. Ξέρουμε, όμως, ότι η διαφορά ανάμεσα στο άθροισμα και στη λίστα είναι η διαφορετική επικεφαλίδα, επομένως αν αλλάξουμε την επικεφαλίδα αυτή μπορούμε να πάρουμε το άθροισμα που θέλουμε:

```
Plus @@ %
```

$$3 - x - 5\,x^2 - 5\,x^3 - 4\,x^4 + 4\,x^5 - 2\,x^6$$

Ο συμβολισμός @@ είναι συντομογραφία της συνάρτησης **Apply**, η οποία όπως έχουμε δει αλλάζει την επικεφαλίδα παραστάσεων:

```
Alias@"@@"D
```

```
Apply
```

Επομένως, βρήκαμε μια μηχανή παραγωγής πολυωνύμων. Μάλιστα θα μπορούσαμε να τη γράψουμε με μορφή συνάρτησης για να μπορούμε να τη χρησιμοποιούμε:

```
randomPoly@n_D :=
 Plus @@ Table@Random@Integer, 8-5, 5<D x^k, 8k, 0, n<D
```

Είναι προφανές, ότι η συνάρτηση αυτή δίνει ένα πολυώνυμο n-ου βαθμού με ακέραιους συντελεστές μεταξύ -5 και 5:

```
p2 = randomPoly@4D
```

$$5\,x^2 - 3\,x^3 - 5\,x^4$$

Επειδή, ο βαθμός ενός πολυωνύμου είναι πάντοτε φυσικός αριθμός, η συνάρτηση αυτή θα μπορούσε να βελτιωθεί, αν επιβάλλουμε το n να είναι ένας φυσικός αριθμός:

```
randomPolynomial@n_Integer?PositiveD :=
 Plus @@ Table@Random@Integer, 8-5, 5<D x^k, 8k, 0, n<D
```

Η επικεφαλίδα Integer αναγκάζει το n να είναι ακέραιος, ενώ ο έλεγχος ?Positive αναγκάζει το n να είναι και θετικός, αν θέλουμε η συνάρτηση randomPolynomial να δώσει απάντησει:

```
p3 = randomPolynomial@5D
```

$$2 - 4\,x - 5\,x^2 + 2\,x^3 - x^4 - 4\,x^5$$

```
randomPolynomial@-5D
```

```
randomPolynomial@-5D
```

```
randomPolynomial@0D

randomPolynomial@0D
```

Τώρα, το *Mathematica* γνωρίζει τρία διαφορετικά πολυώνυμα p1, p2 και p3, των οποίων μπορούμε να υπολογίσουμε το γινόμενό τους:

```
p1  p2  p3

H5 x^2 − 3 x^3 − 5 x^4L H2 − 4 x − 5 x^2 + 2 x^3 − x^4 − 4 x^5L
 H6 + 5 x − 9 x^2 − 3 x^3 + 10 x^4 − 4 x^6L
```

Παρατηρούμε ότι το αποτέλεσμα που επιστρέφει το *Mathematica* δεν είναι ικανοποιητικό, αφού παρουσιάζει το γινόμενο τοποθετώντας το κάθε πολυώνυμο μέσα σε παρενθέσεις.

Το *Mathematica,* όμως, διαθέτει τη συνάρτηση **Expand** με την οποία μπορούμε να αναπτύσουμε ένα γινόμενο.

Expand[expr] αναπτύσσει όλα τα γινόμενα και τις ακέραιες δυνάμεις που υπάρχουν στην παράσταση expr.

```
Expand@%D

60 x^2 − 106 x^3 − 358 x^4 + 359 x^5 + 694 x^6 − 688 x^7 − 564 x^8 +
 960 x^9 + 260 x^10 − 681 x^11 + 54 x^12 + 308 x^13 − 68 x^14 − 80 x^15
```

Επίσης, το *Mathematica* διαθέτει συναρτήσεις με τις οποίες αναλύει πολυώνυμα σε γινόμενο πολυωνύμων:

Factor[poly] μετατρέπει το πολυώνυμο poly σε γινόμενο πολυωνύμων με ακέραιους συντελεστές εφόσον αυτό είναι δυνατό.

Simplify[expr] εκτελεί μια σειρά μετασχηματισμών στην παράσταση expr και επιστρέφει την απλούστερη μορφή που βρίσκει.

Έστω το πολυώνυμο poly:

```
poly = H2 + xL^4 H1 + xL^4 H3 + xL^3;
```

Αναπτύσουμε το γινόμενο και πέρνουμε το πολυώνυμο poly1:

```
poly1 = Expand@polyD

432 + 3024 x + 9432 x^2 + 17296 x^3 + 20715 x^4 +
 17015 x^5 + 9783 x^6 + 3939 x^7 + 1089 x^8 + 197 x^9 + 21 x^10 + x^11
```

Μετρατρέπουμε το πολυώνυμο poly1 σε απλούστερη μορφή:

```
Simplify@poly1D

H3 + xL^3 H2 + 3 x + x^2L^4
```

Αναλύουμε το πολυώνυμο poly1 σε γινόμενο πολυωνύμων:

```
Factor@poly1D

H1 + xL^4 H2 + xL^4 H3 + xL^3
```

Δύο συνήθη προβλήματα που σχετίζονται με τα πολυώνυμα είναι η εύρεση της τιμής ενός πολυωνύμου, όταν η μεταβλητή του αντικαθίσταται με ένα συγκεκριμένο αριθμό και η εύρεση ριζών, εφόσον υπάρχουν. Το *Mathematica* διαθέτει συναρτήσεις, οι οποίες δίνουν απάντηση και στα δύο αυτά προβλήματα, άλλοτε με ακρίβεια και άλλοτε με προσέγγιση.

ReplaceAll[expr, rules] ή expr /. rules	χρησιμοποιεί του κανόνες rules που δίνονται, για να μετασχηματίσει την παράσταση expr.
Solve[eqn, var] var.	επιχειρεί να βρει τις ακριβείς ρίζες της εξίσωσης eqn ως προς τη μεταβλητή

Οι κανόνες (rules) δίνονται με τη μορφή x **Ø** a, όπου το σύμβολο **Ø** είναι ο συνδυασμός των χαρακτήρων - και >.

```
ReplaceAll@poly, x → 1D

82944
```

```
Alias@"ê."D

ReplaceAll
```

Τα πολυώνυμα poly και poly1, τα οποία ορίσαμε προηγουμένως, παριστάνουν ουσιαστικά το ίδιο πολυώνυμο. Επομένως, το Mathematica θα πρέπει να απαντάει καταφατικά στο επόμενο ερώτημα:

```
ReplaceAll@poly, x → 1D « poly1 ê. x → 1

True
```

Η συνάρτηση **Solve** είναι αποτελεσματική, όταν η πολυωνιμική εξίσωση είναι βαθμού το πολύ 4. Ας δούμε μερικά παραδείγματα:

```
Solve@−2 x^2 + 28 x − 39 « 0, xD

        1         è!!!!!!!          1         è!!!!!!!
99x → ¡¡¡ I14 − 118 M=, 9x → ¡¡¡ I14 + 118 M==
        2                           2
```

Παρατηρούμε ότι το αποτέλεσμα της συνάρτησης **Solve** δεν είναι της μορφής x = x_1, αλλά δίνεται σαν κανόνας αντικατάστασης x **Ø** x_1. Αυτό μας επιτρέπει, όπως θα δούμε παρακάτω, να κάνουμε την επαλήθευση.

```
Solve@x^3 − 12 x^2 + 31 x − 27 « 0, xD

                        è!!!!!!!!!!!  1ê3        1        è!!!!!!!!!!!  1ê3
            1  i 837   3 18885  y        I   ¡¡ I279 +  18885 MM
99x → 4 + ¡¡¡ j ¡¡¡¡¡¡ − ¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡ z    + ¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡ =,
            3  k  2          2         {           32ê3
                         è!!!     i 837   3 è!!!!!!!!!!!  y1ê3
 9x → 4 − ¡¡¡ I1 + ą   3 M j ¡¡¡¡¡¡ − ¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡ z    −
           6                k  2          2    18885   {
             è!!!       1           è!!!!!!!!!!!  1ê3
    I1 − ą   3 M I ¡¡ I279 +  18885 MM
    ¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡ =,
                        2 32ê3
            1            è!!!     i 837   3 è!!!!!!!!!!!  y1ê3
 9x → 4 − ¡¡¡ I1 − ą   3 M j ¡¡¡¡¡¡ − ¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡ z    −
           6                k  2          2    18885   {
             è!!!       1           è!!!!!!!!!!!  1ê3
    I1 + ą   3 M I ¡¡ I279 +  18885 MM
    ¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡¡ ==
                        2 32ê3
```

Επίσης, παρατηρούμε ότι μερικές φορές η απάντηση που δίνει η συνάρτηση **Solve** δεν είναι ικανοποιητική. Στις περιπτώσεις αυτές μπορούμε να βρούμε την αριθμητική τιμή των ριζών με δύο τρόπους:

Ο ένας είναι να χρησιμοποιήσουμε τη συνάρτηση **N**:

```
N@%D

88x → 8.83812<, 8x → 1.58094 + 0.745374 ą<,
 8x → 1.58094 − 0.745374 ą<<
```

Ο άλλος είναι να δώσουμε έναν τουλάχιστον από τους συντελεστές του πολυωνύμου με μορφή δεκαδικού αριθμού:

```
Solve@x^3 - 12 x^2 + 31 x - 27.0 « 0, xD

88x → 1.58094 - 0.745374 ą<,
 8x → 1.58094 + 0.745374 ą<, 8x → 8.83812<<
```

Ας δούμε στη συνέχεια πως μπορούμε να κάνουμε την επαλήθευση των ριζών μιας εξίσωσης.

Έστω το πολυώνυμο 3ου βαθμού:

```
a = 6 - 5 x - 2 x^2 + x^3

6 - 5 x - 2 x^2 + x^3
```

Υπολογισμός των ριζών με χρήση της συνάρτησης **Solve**:

```
t = Solve@a « 0, xD

88x → -2<, 8x → 1<, 8x → 3<<
```

Επαλήθευση των ριζών της εξίσωσης με χρήση του τελεστή αντικατάσταση /.:

```
a ê. t

80, 0, 0<
```

Επαλήθευση των ριζών της εξίσωσης με χρήση της συνάρτησης **ReplaceAll**:

```
ReplaceAll@a, tD

80, 0, 0<
```

Όταν έχουμε μια εξίσωση 3ου ή 4ου βαθμού, οι ρίζες δίνονται συνήθως με πολύπλοκες παραστάσεις, οι οποίες τις περισσότερες φορές περιέχουν διπλά ριζικά. Ρίζες με τέτοια μορφή δεν είναι συνήθως χρήσιμες, οπότε σε αυτές τις περιπτώσεις βρίσκουμε τις αριθμητικές τιμές των ριζών είτε με τη χρήση της συνάρτησης **N** είτε με το να δίνουμε έναν συντελεστή της εξίσωσης με μορφή δεκαδικού αριθμού. Όμως, και στις δύο περιπτώσεις, οι ρίζες που θα πάρουμε είναι προσεγγίσεις και όχι ακριβείς ρίζες.

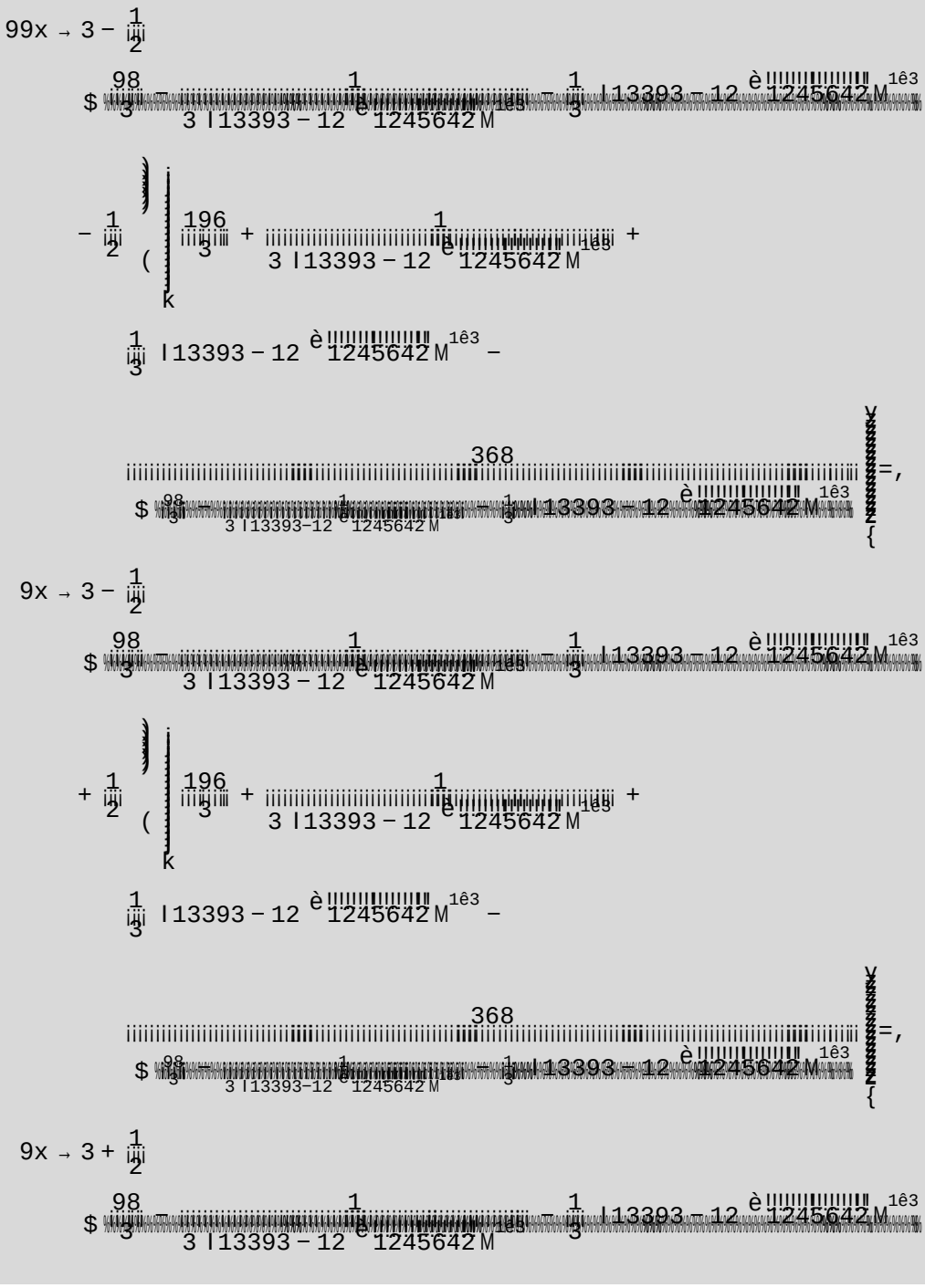
Έστω το πολυώνυμο 4ου βαθμού:

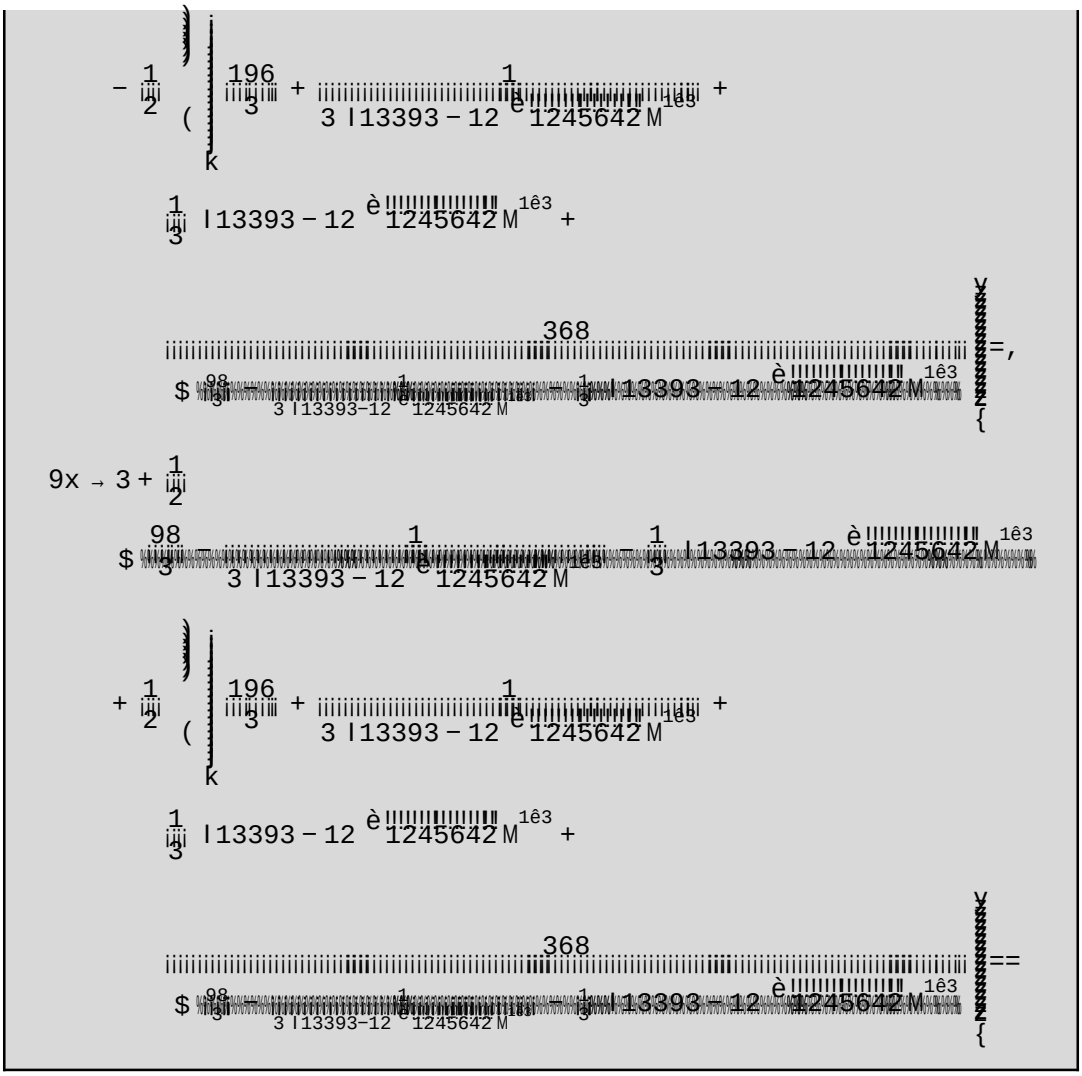
```
b = -8 + 2 x + 5 x^2 - 12 x^3 + x^4
```

$-8 + 2\,x + 5\,x^2 - 12\,x^3 + x^4$

Υπολογισμός των ριζών με χρήση της συνάρτησης **Solve**:

```
Solve@b « 0, xD
```





Υπολογισμός των αριθμητικών τιμών των ριζών με τη χρήση της συνάρτησης **N**:

```
N@%D

88x → 0.618718 - 0.698448 ą<,
 8x → 0.618718 + 0.698448 ą<, 8x → -0.795029<, 8x → 11.5576<<
```

Το *Mathematica* διαθέτει τη συνάρτηση NSolve, με την οποία βρίσκει τις αριθμητικές (προσεγγιστικές) λύσεις μιας εξίσωσεις:

NSolve[eqn, x]	επιστρέφει αριθμητικές λύσεις της εξίσωσης eqn.

Έτσι στην περίπτωση που δεν μας ενδιαφέρουν οι ακριβείς λύσεις, τότε μπορούμε να εφαμόσουμε απ' ευθείας την συνάρτηση **NSolve** για να πάρουμε τις αριθμητικές (προσεγγιστικές λύσεις) λύσεις της εξίσωσης:

```
s = NSolve@b « 0, xD

88x → −0.795029<, 8x → 0.618718 − 0.698448 ą<,
 8x → 0.618718 + 0.698448 ą<, 8x → 11.5576<<
```

Επαλήθευση των ριζών της εξίσωσης με χρήση του τελεστή αντικατάσταση /.:

```
b ê. s

8−4.996 × 10^−16, 3.33067 × 10^−16 − 1.30451 × 10^−15 ą,
 3.33067 × 10^−16 + 1.30451 × 10^−15 ą, 0.<
```

Παρατηρούμε ότι οι τιμές του πολυωνύμου b στις ρίζες s είναι πολύ μικρές, γεγονός που δείχνει την ποιότητα της προσέγγισης, όμως δεν είναι μηδέν, δηλαδή δεν έχουμε ακριβείς ρίζες.

Όταν έχουμε μια εξίσωση 5ου βαθμού και άνω, συνήθως η συνάρτηση **Solve** δεν δίνει ικανοποιητική απάντηση.

Έστω το πολυώνυμο q:

```
q = 1 − 5 x + 7 x^2 + 10 x^3 − 2 x^4 − x^5

1 − 5 x + 7 x^2 + 10 x^3 − 2 x^4 − x^5
```

Υπολογισμός των ριζών με χρήση της συνάρτησης **Solve**:

```
Solve@q « 0, xD

88x → Root@−1 + 5 #1 − 7 #1^2 − 10 #1^3 + 2 #1^4 + #1^5 &, 1D<,
 8x → Root@−1 + 5 #1 − 7 #1^2 − 10 #1^3 + 2 #1^4 + #1^5 &, 2D<,
 8x → Root@−1 + 5 #1 − 7 #1^2 − 10 #1^3 + 2 #1^4 + #1^5 &, 3D<,
 8x → Root@−1 + 5 #1 − 7 #1^2 − 10 #1^3 + 2 #1^4 + #1^5 &, 4D<,
 8x → Root@−1 + 5 #1 − 7 #1^2 − 10 #1^3 + 2 #1^4 + #1^5 &, 5D<<
```

Παρατηρούμε ότι η συνάρτηση **Solve** μας δίνει 5 ρίζες, όμως η μορφή τους δεν είναι αυτή που θέλαμε. Σε αυτές τις περιπτώσεις μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **NSolve** για να πάρουμε τις αριθμητικές (προσεγγιστικές) τιμές των ριζών:

```
NSolve@q « 0, xD

88x → −3.98547<, 8x → −1.11737<, 8x → 0.248221 − 0.156657 ą<,
 8x → 0.248221 + 0.156657 ą<, 8x → 2.6064<<
```

```
Clear@a, bD
```

Εκτός από τα πολυώνυμα, το *Mathematica* μπορεί να διαχειριστεί και ρητές συναρτήσεις. Άλλωστε οι ρητές συναρτήσεις είναι, όπως ξέρουμε, πηλίκο δύο πολυωνύμων, επομένως συνδέονται αρκετά στενα με τα πολυώνυμα. Συνήθη προβλήματα που αφορούν τις ρητές συναρτήσεις είναι η απλοποίηση κοινών παραγόντων από αριθμητή και παρανομαστή, ή η ανάλυση σε απλούστερα κλάσματα.

Το *Mathematica* διαθέτει αρκετές συναρτήσεις οι οποίες αντιμετωπίζουν αυτά τα συνήθη προβλήματα. Μερικές από αυτές είναι:

Apart[expr] αναλύει τη ρητή παράσταση expr σε απλούστερα κατά το δυνατόν κλάσματα.

```
Apart@1 ê H3 + 10 x + 12 x^2 + 6 x^3 + x^4LD

      1          1          1          1
- ──────── + ──────── + ──────── - ────────
  2 H1 + xL3   4 H1 + xL2   8 H1 + xL   8 H3 + xL
```

```
Apart@H2 − 3 x^2L ê H2 − 3 x + 2 x^2 − 2 x^3 + x^5LD

      1          7          2        1 + 2 x
- ──────── - ──────── - ──────── + ──────────
  6 H−1 + xL2   9 H−1 + xL   9 H2 + xL   2 H1 + x2L
```

Apart[expr, var] το ίδιο με την συνάρτηση Apart[expr], με τη διαφορά ότι χρησιμοποιούνται μόνον οι μεταβλητές var, ενώ οι υπόλοιπες θεωρούνται ως σταθερές.

```
Apart@Hx − yL ê H6 + 5 x − 2 x^2 − x^3LD

      x                y
- ──────────── + ────────────
  −6 − 5 x + 2 x2 + x3   −6 − 5 x + 2 x2 + x3
```

```
Apart[x - y] ê [6 + 5 x - 2 x^2 - x^3, x]
```

$$\frac{-1-y}{6\,(1+x)} + \frac{-2+y}{15\,(-2+x)} + \frac{3+y}{10\,(3+x)}$$

Together[expr] προσθέτει τους όρους της παράστασης expr και στο αποτέλεσμα επιχειρεί να απλοποιήσει κοινούς παράγοντες από αριθμητή και παρανομαστή.

```
Together[a ê b + x ê y]
```

$$\frac{b\,x + a\,y}{b\,y}$$

```
Together[(1 + Sqrt[2]) ê (a - Sqrt[2]) - (1 - Sqrt[2]) ê (a + Sqrt[2])]
```

$$-\frac{2\,\left(\sqrt{2} + \sqrt{2}\right)}{\left(a - \sqrt{2}\right)\left(a + \sqrt{2}\right)}$$

Collect[expr, x] από τους όρους της παράστασης expr βγάζει κοινό παράγοντα τις δυνάμεις του x.

```
Collect[x^4 + (a + y) (x + y^3) x^2 + (y^3 + 3) x^2, x]
```

$$x^4 + x^3\,(a + y) + x^2\,(3 + y^3 + y^3\,(a + y))$$

```
Collect[x^4 + (a + y) (x + y^3) x^2 + (y^3 + 3) x^2, y]
```

$$3\,x^2 + a\,x^3 + x^4 + x^3\,y + (x^2 + a\,x^2)\,y^3 + x^2\,y^4$$

```
Collect[x^4 + (a + y) (x + y^3) x^2 + (y^3 + 3) x^2, {x, y}]
```

$$x^4 + x^3\,(a + y) + x^2\,(3 + (1 + a)\,y^3 + y^4)$$

Cancel[expr] απλοποιεί κοινούς παράγοντες από αριθμητή και παρανομαστή στην παράσταση expr.

```
Cancel@Hx^3 - 4 xL ê Hx^3 - 2 x^2L + Hx^2 - 3 x + 2L ê Hx^2 - 1LD
```

$$\frac{-2+x}{1+x}+\frac{2+x}{x}$$

```
Together@Hx^3 - 4 xL ê Hx^3 - 2 x^2L + Hx^2 - 3 x + 2L ê Hx^2 - 1LD
```

$$\frac{2+x+2x^2}{x(1+x)}$$

Κεφάλαιο 3ο: Οι φυσικοί και οι ακέραιοι αριθμοί

3.1 Οι αριθμοί στο *Mathematica*

Το *Mathematica* δεν αναγνωρίζει το 7. σαν κάποιο ακέραιο αριθμό. Με την Head μπορούμε να δούμε την επικεφαλίδα ενός αριθμού.

```
Head@7.D

Real
```

Υπάρχουν πολλές άλλες δυνατότητες για αριθμούς όπως Integer, Rational, Complex π.χ

```
9Head@3D, HeadA 1
                ijjj E, Head@0.3D, Head@0.3 + ąD=
                3

8Integer, Rational, Real, Complex<
```

Οι πράξεις εκτελούνται με διαφορετικό τρόπο σε κάθε περίπτωση. Έτσι όταν δουλεύομαι με ακέραιους αριθμούς πρέπει κάθε φορά να σιγουρευόμαστε ότι οι μεταβλητές μας και οι σταθερές μας είναι ακέραιοι αριθμοί στις εμπλεκόμενες σχέσεις π.χ όταν θελήσουμε να ορίσουμε το παραγοντικό για ακεραίους θα γράψουμε:

```
paragontiko@n_IntegerD := HnL !
```

Με το n_Integer αναγκάζουμε την μεταβλητή να παίρνει ακέραιες μόνο τιμές-αλλιώς δεν θ προκύψει αποτέλεσμα.

```
8paragontiko@4D, paragontiko@4.D, H4.5L !<

8paragontiko@4D, paragontiko@4.D, 52.3428<
```

3.2 Οι Διαιρέτες ενός ακεραίου αριθμού

Η συνάρτηση Divisors βρίσκει όλους τους θετικούς διαιρέτες ενός ακεραίου.

```
Divisors@-72D

81, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72<
```

Αν θέλουμε μόνο τους πρώτους διαιρέτες απ' αυτούς τότε γράφουμε

```
Select@%, PrimeQD

82, 3<
```

Με το PrimeQ ελέγχουμε αν ένας αριθμός είναι πρώτος ή όχι:

```
8PrimeQ@12D, PrimeQ@-3D, PrimeQ@7.D<

8False, True, False<
```

Απο την απάντηση βλέπουμε ότι το 7. και το 12 δεν είναι πρώτοι ενώ το -3 θεωρείται πρώτος!

3.3 Παραγοντοποίηση ενός ακεραίου αριθμού

Αν θέλουμε τους πρώτους παράγοντες μαζί με τους εκθέτες τους στο ανάπτυγμα ενός ακεραίου τότε γράφουμε

```
FactorInteger@140D

882, 2<, 85, 1<, 87, 1<<
```

Το {2,2) σημαίνει 2^2 και όμοια το 85, 1< σημαίνει 5^1 στην ανάλυση του 140. Για αρνητικούς παίρνουμε μπροστά και το {-1,1} για παράδειγμα

```
FactorInteger@-140D

88-1, 1<, 82, 2<, 85, 1<, 87, 1<<
```

Τώρα θα ορίσουμε μια συνάρτηση ώστε να εμφανίζεται μονάχα ο μέγιστος πρώτος διαιρέτης ενός ακεραίου. Για τον σκοπό αυτό κατασκευάζουμε την maxPrimeDivisor

```
maxPrimeDivisor@x_IntegerD := First@Last@FactorInteger@xDDD
```

```
maxPrimeDivisor@14D

7
```

Η χρήση των First και Last φαίνεται στα παραδείγματα που ακολουθούν.

```
u = 81, 5, 7, 7, 8, -1<

81, 5, 7, 7, 8, -1<
```

```
Last@uDH  το πρώτο από δεξιά L

-1
```

```
First@uDH  το πρώτο από αριστερά L

1
```

3.4 Οι συναρτήσεις Transpose και Part

Ένας άλλος τρόπος για την εύρεση των πρώτων διαιρετών θα ήταν να απομονώσουμε τους πρώτους που βρίσκονται μέσα στην λίστα FactorInteger. Αυτό μπορεί να γίνει είτε με αναστροφή(Transpose) του πίνακα FactorInteger είτε διαλέγοντας μόνο τις πρώτες συντεταγμένες από κάθε ζευγάρι του FactorInteger με την χρήση της Part.

```
? Part
```

```
expr@@iDD or Part@expr, iD gives the ith part of expr.
  expr@@-iDD counts from the end. expr@@0DD gives the head
  of expr. expr@@i, j, ... DD or Part@expr, i, j, ... D
  is equivalent to expr@@iDD @@jDD @@ DD... . expr@@ 8i1,
  i2, ... < DD gives a list of the parts i1, i2, ...  of expr.
```

Με άλλα λόγια η Part[expr,i] δίνει το i μέρος της expr ενώ με Part[expr,i,j] παίρνουμε το j μέρος του i.

```
m = FactorInteger@15D
MatrixForm@mDH  Ο m σε μορφή πίνακα L
anastrofos = Transpose@mDH αναστροφή του m  L
First@anastrofosDH Παίρνουμε την 1 η γραμμή  L


883, 1<, 85, 1<<
```

$$\begin{pmatrix} 3 & 1 \\ 5 & 1 \end{pmatrix}$$

```
883, 5<, 81, 1<<
```

```
83, 5<
```

Άλλος τρόπος με την χρήση της Part

```
Part@m, 81<DH  η πρώτη γραμμή του m  L
Part@m, 81<, 81<D
H  το στοιχείο της 1 ης γραμμής, 1 ης στήλης  L
Part@m, All, 81<DH  η 1 η στήλη του m−
  αυτή φυσικά περιέχει τους πρώτους που μας ενδιαφέρουν  L


883, 1<<
```

```
883<<
```

```
883<, 85<<
```

Άρα οι πρώτοι διαιρέτες είναι το 3 και το 5. Αν δεν μας αρέσουν οι πολλές αγκύλες μπορούμε να τις απλοποιήσουμε με την χρήση της Flatten.

```
Flatten@%D

83, 5<
```

Όλες οι παραπάνω πράξεις μπορούν να γραφούν με μία μόνο συνάρτηση:

```
diairetes@n_IntegerD := Flatten@Part@FactorInteger@nD, All, 81<DD
```

3.5 Οι πρώτοι αριθμοί

Αν ζητάμε να βρούμε τους πρώτους<=10 τότε

```
Select@Range@10D, PrimeQD

82, 3, 5, 7<
```

ενώ αν ζητάμε τους 10 πρώτους στη σειρά τότε

```
Prime@Range@10DD

82, 3, 5, 7, 11, 13, 17, 19, 23, 29<
```

Η συνάρτηση Prime[x] παραπάνω μας δίνει το x -ιοστό πρώτο. Με ?Prime μπορούμε να μάθουμε πληροφορίες.

```
Prime@300D

1987
```

```
? Prime
```

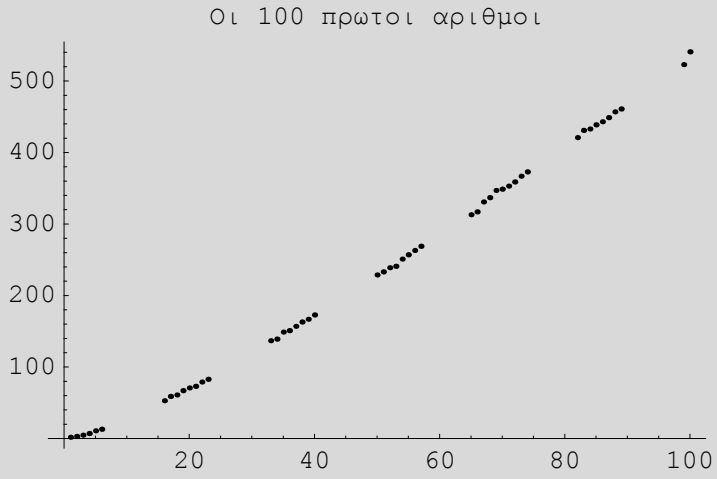
```
Prime@nD gives the nth prime number.
```

Οι 100 πρώτοι αριθμοί δίνονται με την μορφή πίνακα.

```
first100prwtoi = Table@8n, Prime@nD<, 8n, 100<D

881, 2<, 82, 3<, 83, 5<, 84, 7<, 85, 11<, 86, 13<, 87, 17<, 88, 19<,
 89, 23<, 810, 29<, 811, 31<, 812, 37<, 813, 41<, 814, 43<,
 815, 47<, 816, 53<, 817, 59<, 818, 61<, 819, 67<, 820, 71<,
 821, 73<, 822, 79<, 823, 83<, 824, 89<, 825, 97<, 826, 101<,
 827, 103<, 828, 107<, 829, 109<, 830, 113<, 831, 127<, 832, 131<,
 833, 137<, 834, 139<, 835, 149<, 836, 151<, 837, 157<, 838, 163<,
 839, 167<, 840, 173<, 841, 179<, 842, 181<, 843, 191<, 844, 193<,
 845, 197<, 846, 199<, 847, 211<, 848, 223<, 849, 227<,
 850, 229<, 851, 233<, 852, 239<, 853, 241<, 854, 251<, 855, 257<,
 856, 263<, 857, 269<, 858, 271<, 859, 277<, 860, 281<,
 861, 283<, 862, 293<, 863, 307<, 864, 311<, 865, 313<,
 866, 317<, 867, 331<, 868, 337<, 869, 347<, 870, 349<, 871, 353<,
 872, 359<, 873, 367<, 874, 373<, 875, 379<, 876, 383<, 877, 389<,
 878, 397<, 879, 401<, 880, 409<, 881, 419<, 882, 421<, 883, 431<,
 884, 433<, 885, 439<, 886, 443<, 887, 449<, 888, 457<, 889, 461<,
 890, 463<, 891, 467<, 892, 479<, 893, 487<, 894, 491<, 895, 499<,
 896, 503<, 897, 509<, 898, 521<, 899, 523<, 8100, 541<<
```

Η γραφική παράσταση των παραπάνω σημείων είναι:

```
ListPlot@first100prwtoi, PlotLabel -> "Οι 100 πρωτοι αριθμοι"D
```




```
¦ Graphics ¦
```

3.6 Η Ευκλείδια Διαίρεση

Είναι γνωστό ότι αν α, β είναι ακέραιοι με β μη μηδενικός τότε υπάρχουν μοναδικοί ακέραιοι q=Quotient[α,β](το πηλίκο της διαίρεσης) m=Mod[α,β] (το υπόλοιπο) έτσι ώστε α=qβ+m. Το q θα μπορούσε να γραφεί συναρτήσει των α,β ως εξής q=sgnβ μ A$\frac{\alpha}{\beta}$E όπου με sgnβ συμβολίζουμε το πρόσημο $\frac{\alpha}{\beta}$του β και με [...] το ακέραιο μέρος ενός πραγματικού αριθμού. Βέβαια το ακέραιο μέρος στο *Mathematica* έχει την έννοια του μέρους που βρίσκεται αριστερά απο το δεκαδικό μέρος του π.χ

```
IntegerPart@-PiD

-3
```

Ας μελετήσουμε λίγο την συνάρτηση Mod[α,β]. Ως το υπόλοιπο της διαίρεσης του α με το β πρέπει να είναι πάντα θετικό ή 0 και μικρότερο της απόλυτης τιμής του β.

```
Mod@-31, 5D

4
```

Το πηλίκο όμως μπορεί να είναι αρνητικό.

```
Quotient@-31, 5D

-7
```

Φυσικά, τα νούμερα που βρήκαμε επαληθεύουν την ταυτότητα της διαίρεσης.

```
-31 « 5 Quotient@-31, 5D + Mod@-31, 5D

True
```

Γενικά το *Mathematica* μας δίνει δυνατότητες να ελένξουμε(για απλές τουλάχιστον περιπτώσεις) αν ισχύουν κάποιες ταυτότητες π.χ

```
Sum@k^2, 8k, 1, n<D « n Hn + 1L H2 n - 1L ê 6

1/6 n H1 + nL H1 + 2 nL == 1/6 n H1 + nL H-1 + 2 nL
```

Δεν έβγαλε true. Άρα μάλλον εμείς έχουμε γράψει στο δεξιό σκέλος της παραπάνω ισότητας λανθασμένα ή το *Mathematica* αδυνατεί να μας το αποδείξει . Ας το διορθώσουμε το 2n-1 σε 2n+1:

```
Sum@k^2, 8k, 1, n<D « n Hn + 1L H2 n + 1L ê 6

True
```

Θα πρέπει να θυμούμαστε πάντα ότι το *Mathematica* μπορεί να κάνει πράξεις και να χειρίζεται σύμβολα αλλά δεν μπορεί να κάνει μαθηματικά δηλ. δεν μπορεί να μας αποδείξει μια ταυτότητα π.χ με επαγωγή. Απλώς μπορούμε να διαπιστώσουμε αν πράγματι ισχύει η ταυτότητα δοκιμάζοντας την με κάποιους αριθμούς n χωρίς βέβαια αυτό να αποτελεί απόδειξη. Π.χ για n απο ένα μέχρι το 10 βλέπουμε ότι η παραπάνω ταυτότητα αληθεύει

```
Table@Sum@k^2, 8k, 1, n<D « n Hn + 1L H2 n + 1L ê 6, 8n, 1, 10<D

8True, True, True, True, True, True, True, True, True, True<
```

3.7 Η εικασία του Goldbach

Το 1742 ο Christian Goldbach διατύπωσε την εικασία ότι κάθε άρτιος θετικός ακέραιος μεγαλύτερος του 4 μπορεί να γραφεί σαν άθροισμα δύο πρώτων αριθμών. Το *Mathematica* δεν μπορεί βέβαια να δώσει την απόδειξη της εικασίας αυτής, μπορεί όμως να μας βοηθήσει να βρούμε ζεύγη πρώτων αριθμών που το άθροισμα τους είναι δεδομένο. Παρακάτω ορίζουμε την συνάρτηση goldbach για αυτόν το σκοπό. Θα πρέπει να αναφέρουμε ότι δεξιά του /; μπαίνουν οι συνθήκες που πρέπει οποσδήποτε να ικανοποιούνται. Με EvenQ[n] ελέγχουμε αν ο n είναι άρτιος και με Positive[n] αν είναι θετικός. Για την συνάρτηση For θα πούμε αργότερα κάποια πράγματα. Για τώρα μπορείται να μάθετε πληροφορίες με το ?For.

```
goldbach@n_Integer ê; EvenQ@nD && Positive@nD && n > 4D :=
 Module@8a = 8<, i<,
  For@i = 2, Prime@iD ≤ n ê 2, i++, If@PrimeQ@n - Prime@iDD,
    a = Append@a, 8Prime@iD, n - Prime@iD<DDD; aD
```

```
goldbach@2000D

883, 1997<, 87, 1993<, 813, 1987<, 867, 1933<, 8127, 1873<,
 8139, 1861<, 8199, 1801<, 8211, 1789<, 8223, 1777<, 8241, 1759<,
 8277, 1723<, 8307, 1693<, 8331, 1669<, 8337, 1663<, 8373, 1627<,
 8379, 1621<, 8421, 1579<, 8433, 1567<, 8457, 1543<, 8541, 1459<,
 8547, 1453<, 8571, 1429<, 8577, 1423<, 8601, 1399<, 8619, 1381<,
 8673, 1327<, 8709, 1291<, 8751, 1249<, 8769, 1231<,
 8787, 1213<, 8829, 1171<, 8877, 1123<, 8883, 1117<,
 8907, 1093<, 8937, 1063<, 8967, 1033<, 8991, 1009<<
```

Το a είναι η λίστα που περιέχει όλα τα δυνατά ζεύγη πρώτων στα οποία η 1η συντεταγμένη είναι <= της 2ης. Αρχικά είναι κενή (a={ })και κάθε φορά που βρίσκουμε ένα ζευγάρι πρώτων με άθροισμα n το επισυνάπτουμε με την Append στην a. Το Module δηλώνει ένα σύνολο "τοπικών" (local)εντολών δηλ. που

εκτελούνται μόνο όταν καλείται η συνάρτηση goldbach. Το Module ξεκινάει πάντα με την δήλωση των "τοπικών μεταβλητών" και των αρχικών τιμών που πιθανόν να έχουν. Στο παραπάνω Module οι τοπικές μεταβλητές είναι η a και η i.

Module[{x, y, ...}, expr] specifies that occurrences of the symbols x, y, ... in expr should be treated local. Module[{x = x0, ...}, expr] defines initial values for x, ...

3.8 Ο μέγιστος κοινός διαιρέτης και το ελάχιστο κοινό πολλαπλάσιο

Με GCD παίρνουμε τον μέγιστο κοινό διαιρέτη κάποιων ακεραίων. Π.χ

```
In[]:= {GCD[24, 14, 8], GCD[-24, 14, 8], GCD[24., 14, 8]}
```

```
GCD::exact : Argument 24.` at position 3 is not an exact number.
```

```
Out[]= {2, 2, GCD[8, 14, 24.]}
```

Βλέπουμε ότι το GCD δεν δουλεύει για πραγματικούς και βγάζει μήνυμα λάθους. Το GCD[x,y,z] είναι πάντα ίσο με GCD[GCD[x,y],z] και μπορούμε να το διαπιστώσουμε εύκολα:

```
In[]:= {GCD[x, y, z] « GCD[GCD[x, y], z], GCD[x, y] « GCD[y, Mod[x, y]]}
```

```
Out[]= {True, GCD[x, y] == GCD[y, Mod[x, y]]}
```

Παρατηρήστε ότι για την μια ταυτότητα δίνει True ενώ για την άλλη(που είναι και αυτή αληθινή) δεν δίνει απάντηση!
Για διδακτικούς λόγους θα ορίσουμε τώρα το GCD[x,y,z] ως τον μεγαλύτερο απο τους κοινούς διαρέτες των x,y,z. Ας βρούμε για παράδειγμα το GCD[24,14,8]. Βρίσκουμε πρώτα ποιοί διαιρέτες είναι οι κοινοί χρησιμοποιώντας την συνάρτηση Intersection(κοινή τομή):

```
In[]:= Intersection[Divisors[24], Divisors[14], Divisors[8]]
```

```
Out[]= {1, 2}
```

Στην συνέχεια παίρνουμε το τελευταίο στοιχείο που εμφανίζεται στην τομή. Εδώ όπως βλέπουμε συμπίπτει με το μεγαλύτερο στοιχείο της τομής(δηλ. το 2) που είναι και ο ζητούμενος Μ.Κ.Δ.

```
In[]:= Last[%]
```

```
Out[]= 2
```

Μπορούμε να βάλουμε όλα τα παραπάνω μέσα σε μια και μόνο συνάρτηση:

```
megkoinosDiairetis@m_Integer, n_IntegerD :=
 Last@Intersection@Divisors@mD, Divisors@nDDD
```

```
megkoinosDiairetis@24, 14D

2
```

Μπορεί επίσης να δοθεί και ένας αναδρομικός ορισμός της GCD χρησιμοποιώντας την ταυτότητα:

```
GCD@m, nD « GCD@n, Mod@m, nDD

GCD@m, nD == GCD@n, Mod@m, nDD
```

Ο αναδρομικός ορισμός θα δοθεί μετην myGCD στην επόμενη ενότητα. Όπως ήδη αναφέραμε το *Mathematica* αδυνατεί να αποδείξει την ταυτότητα παρόλο που για κάθε συγκεκριμένες τιμές ισχύει π.χ.

```
GCD@24, 14D == GCD@24, Mod@24, 14DD

True
```

Το ελάχιστο κοινό πολλαπλάσιο κάποιων αριθμών βρίσκεται με το LCM.

```
LCM@-2, -3, -5D

30
```

Φυσικά μπορούμε να το ορίσουμε χρησιμοποιώντας το GCD πού ήδη είδαμε ως εξής:

```
myLCM@m_, n_D := Abs@m  nD ê GCD@m, nD
```

```
myLCM@-2, -3D

6
```

Το Abs δίνει την απόλυτη τιμή.

3.9 Αναδρομικοί ορισμοί στο *Mathematica*

Κάθε αναδρομικός ορισμός αποτελείται απο τις αρχικές(οριακές)συνθήκες και απο το κυρίως σώμα. Π.χ

```
myGCD@m_, 0D := m
myGCD@m_, n_D := myGCD@n, Mod@m, nDD
```

```
myGCD@24, 14D

2
```

Χρειάζεται ιδιαίτερη προσοχή όταν ορίζουμε αναδρομές. Έτσι για παράδειγμα αν θα τρέξουμε το myGCD[4.,2] θα πέσουμε σε ατέρμονα loop!(Φταίει ότι το 4. δεν είναι ακέραιος. Για να σταματήσει να τρέχει το πρόγραμμα πατάμε Alt +) Θα έπρεπε τα m,n να δηλωθούν ως ακέραιοι: myGCD[m_Integer,0]:=m και myGCD[m_Integer,n_Integer]:=
myGCD[n,Mod[m,n]]. Ένας ισοδύναμος τρόπος για να δηλωθούν τα m και n ως ακέραιοι είναι ο παρακάτω

```
Clear@myGCDD
myGCD@m_, 0D ê; IntegerQ@mD := m
myGCD@m_, n_D ê; IntegerQ@nD && IntegerQ@mD :=
 myGCD@n, Mod@m, nDD
```

```
8myGCD@24, 14D, myGCD@24., 14D<

82, myGCD@24., 14D<
```

Το /; δηλώνει τις συνθήκες που πρέπει να ικανοποιούνται απαραίτητα για να ισχύει ο ορισμός. Το IntegerQ[n] δίνει true αν το n είναι ακέραιος. Το && είναι το σύμβολο της σύζευξης στο *Mathematica*. Το myGCD[24.,14]δεν υπολογίσθηκε διότι το 24. είναι Real.

Ένα άλλο παράδειγμα αναδρομικού ορισμού μπορεί να δοθεί για την κατασκευή μιας ακολουθίας. Π.χ η ακολουθία 1,1,2,3,5,.. που κάθε όρος της είναι το άθροισμα των δύο προηγούμενων λέγεται ακολουθία του Fibonacci και δίνεται στο *Mathematica* με την χρήση της ενσωματωμένης συνάρτησης Fibonacci π.χ.

```
8Fibonacci@1D, Fibonacci@2D,
 Fibonacci@3D, Fibonacci@4D, Fibonacci@5D<

81, 1, 2, 3, 5<
```

Ένας αναδρομικός ορισμός θα μπορούσε να είναι ο

```
Remove@myFibD
myFib@1D := 1;H   πρώτη αρχική συνθήκη  L
myFib@2D := 1;  H δεύτερη αρχική συνθήκη  L
myFib@n_IntegerD := myFib@n - 1D + myFib@n - 2D
```

```
myFib@4D

3
```

Ο παραπάνω αναδρομικός ορισμός δεν είναι και ιδιαίτερα καλός διότι θα αναγκαστεί ο υπολογιστής να κρατησει πολλές φορές κάθε προηγούμενο όρο. Αυτό το διαπιστώνουμε χρησιμοποιώντας την εντολή Trace.

```
Trace@myFib@5DD

8myFib@5D, myFib@5 - 1D + myFib@5 - 2D,
 885 - 1, 4<, myFib@4D, myFib@4 - 1D + myFib@4 - 2D,
  884 - 1, 3<, myFib@3D, myFib@3 - 1D + myFib@3 - 2D,
   883 - 1, 2<, myFib@2D, 1<, 883 - 2, 1<, myFib@1D, 1<, 1 + 1, 2<,
  884 - 2, 2<, myFib@2D, 1<, 2 + 1, 3<, 885 - 2, 3<, myFib@3D,
  myFib@3 - 1D + myFib@3 - 2D, 883 - 1, 2<, myFib@2D, 1<,
  883 - 2, 1<, myFib@1D, 1<, 1 + 1, 2<, 3 + 2, 5<
```

Παρατηρούμε ότι το myFib[2] έχει εμφανιστεί 3 φορές στον υπολογισμο του myFib[5] όπως εύκολα βλέπουμε με το Trace[myFib[5],myFib[_]]

```
8Trace@myFib@5D, myFib@_DD,
 TimeConstrained@myFib@200D, 1.5D, Timing@Fibonacci@200DD<

88myFib@5D,
  8myFib@4D, 8myFib@3D, 8myFib@2D<, 8myFib@1D<<, 8myFib@2D<<,
  8myFib@3D, 8myFib@2D<, 8myFib@1D<<<, $Aborted,
 80. Second, 28057117299251014003761193241303867718952 5<<
```

Ακόμα βλέπουμε ότι το myFib[200] είναι αδύνατον να υπολογιστεί σε λιγότερο απο 1.5 δευτερόλεπτα(αυτό σημαίνει το $Aborted) ενώ το Fibonacci[200] υπολογίστηκε σε 0. Second δηλ. σχεδόν σε μηδενικό χρόνο και να φανταστείται ότι το MyFib[200]=280571172992510140037611932413038677189525. Όλα αυτά δείχνουν καθαρά ότι ο ορισμός που δώσαμε δεν είναι τόσο γρήγορος. Για αυτό το λόγο επιλέγουμε ένα καλύτερο ορισμό τον newFib που υπολογίζει την ακολουθία από κάτω πρός τα πάνω δηλ. βρίσκουμε κάθε φορά την επόμενη τιμή ξεκινώντας απο τις αρχικές χωρίς επαναλήψεις υπολογισμών.

```
newFib@1D := 1;
newFib@n_D :=
 Module@8x = 0, y = 1<, Do@8x, y< = 8y, x + y<, 8n - 1<D; yD
```

```
Timing@newFib@200DD

80. Second, 280571172992510140037611932413038677189525<
```

Τελειώνουμε με ένα παράδειγμα **κακού** αναδρομικού ορισμού. Οι κανόνες εκτελούνται με την σειρά που τους δίνομε. Δεν πρέπει να υπάρχουν επικαλύψεις των αρχικών συνθηκών όπως συμβαίνει π.χ με τον υπολογισμό του bad[0,0] παρακάτω ή να υπάρχει loop στο κυρίως σώμα της αναδρομής όπως συμβαίνει π.χ με τον υπολογισμό του badgcd[Mod[9,15],15]] . Ένας κακός αναδρομικός ορισμός σίγουρα δεν θα δουλέψει!

```
badgcd@a_, 0D := a
badgcd@0, b_D := b
badgcd@a_, b_D := badgcd@Mod@a, bD, bD
```

```
badgcd@24, 15D
```

```
$IterationLimit::itlim : Iteration limit of 4096 exceeded.
```

```
Hold@badgcd@Mod@9, 15D, 15DD
```

3.10 Επίλυση Διοφαντικών εξισώσεων

Όταν λέμε διοφαντική εξίσωση εννοούμε μια συνηθισμένη γραμμική εξίσωση όπου οι άγνωστοι μπορούν να παάρουν μόνο ακέραιες λύσεις. Π.χ ax+by=c. Σε αυτέ τις περιπτώσεις για να υπάρχει λύση θα πρέπει απαραίτητα ο Μ.Κ.Δ των συντελεστών των αγνώστων να διαιρεί το c. Δηλ θα πρέπει με ορολογία του *Mathematica* το Mod[c,GCD[a,b]]=0. Π.χ για c=GCD[a,b] η εξίσωση ax+by=GCD[a,b] έχει σίγουρα μια ακέραια λύση {m,n} δηλ. am+bn=GCD[a,b]. Το *Mathematica* μας δίνει την δυνατότητα να βρούμε συγχρόνως το GCD[a,b] και μια ειδική λύση {m,n} της παραπάνω εξίσωσης. Η συνάρτηση για αυτό το σκοπό είναι η ExtendedGCD. π.χ

```
ExtendedGCD@12, 7, 245D

81, 83, -5, 0<<
```

Δηλαδή ΜΚΔ=1 και η διοφαντική εξίσωση 12m+7n+245r=1 έχει μια λύση {m,n,r}={3,-5,0}. Φυσικά ίσως να υπάρχουν και άλλες ακέραιες λύσεις. Με την χρήση της If μπορούμε να τσεκάρουμε αν μια διοφαντική εξίσωση έχει λύση. Ακολουθούν κάποια παραδείγματα.

```
If@Mod@16, GCD@12, 8DD « 0,
 " η εξίσωση έχει ακέραια λύση", "καμμία ακέραια λύση"D

 η εξίσωση έχει ακέραια λύση
```

```
If@Mod@16, GCD@12, 3, 15DD « 0,
 " η εξίσωση έχει ακέραια λύση", "καμμία ακέραια λύση"D

καμμία ακέραια λύση
```

Με άλλα λόγια η 16=12m+8n έχει ακέραιες λύσεις ενώ η 16=12x+3y+15z δεν έχει.
Ειδικά όταν έχουμε δυο αγνώστους - την x και y-τότε μπορούμε εύκολα να βρούμε την γενική λύση ώς εξής: Ορίζουμε κατ'αρχήν d=GCD[a,b]. Έστω ότι d/c και m,n είναι μια ακέραια λύση της am+bn=d. Πολλαπλασιάζουμε και τα δύο μέλη με τον ρητό c/d και παίρνουμε $ax_0 + by_0 = c$ όπου $x_0 = Hc ê dL\, m$ και $y_0 = Hc ê dL\, n$. Τώρα μπορούμε να πάρουμε μια γενική ακέρια λύση σύμφωνα με τον τύπο $x = x_0 + bt$, $y = y_0 - at$ όπου t είναι μια παράμετρος με ακέραιες τιμές. Όλα τα παραπάνω υλοποιούνται ορίζοντας την συνάρτηση diofantine.

```
diofantine@a_Integer, b_Integer, c_IntegerD
 H  λύση της ax+by=c  L := H8d, 8m, n<< = ExtendedGCD@a, bD;
  H Τα m και n ικανοποιούν την  am+bn=
    d όπου d είναι ο ΜΚΔ των a,b  LIf@Mod@c, dD « 0,
   8Hc ê dL m + b t, Hc ê dL n - a t<, "δεν έχει ακέραια λύση"DL
```

```
8diofantine@12, 7, 245D, diofantine@12, 8, 245D<

88735 + 7 t, -1225 - 12 t<, δεν έχει ακέραια λύση<
```

Με άλλα λόγια η 12x+7y=245 έχει λύσεις τις {735+7 t,-1225-12 t} ενώ η 12x+8y=245 δεν έχει.

```
m = diofantine@12, 7, 245D
m = m ê. t → 100;
x = First@mD
y = Last@mD
12 x + 7 y « 245

8735 + 7 t, −1225 − 12 t<
```

```
1435
```

```
−2425
```

```
True
```

Το /. είναι σύμβολο της αντικατάστασης. Με m=m/.tØ100 αντικαταστήσαμε την τμε 100 στο τύπο του m και το αποτέλεσμα (το ζευγάρι 1435, -2425 μπήκε στο m. Στην συνέχεια βρίσκουμε τα αντίστοιχα x,y και κάνουμε την δοκιμή. Το ερωτηματικό ; στην m=m/.tØ100; σημαίνει ότι δεν θέλουμε να "εκτυπώσουμε" το m στην οθόνη μας αλλά απλώς να γίνουν οι πράξεις "σιωπηλά".

3.11 Λύνοντας εξισώσεις με modulus

Στην θεωρία αριθμών συναντάμε εξισώσεις και συστήματα με modulus. Για παράδειγμα 2x-3y+z=11(mod 5). Εδώ ζητάμε ακέραιες λύσεις για τα x,y,z που παίρνουν τιμές 0,1,2,3 και 4. Τέτοιες γραμμικές εξισώσεις αντιμετωπίζονται με την χρήση της LinearSolve ή της Solve όπως βλέπουμε στα παραδείγματα. Αν η εξίσωση ή οι εξισώσεις δεν είναι γραμμικές τότε θα πρέπει να χρησιμοποίσουμε την Solve.

```
H Μια λύση της εξίσωσης 2 x−3 y+z=11 Hmod 5L L
a = 882, −3, 1<<; b = 811<; LinearSolve@a, b, Modulus → 5D

83, 0, 0<
```

```
Solve@2 x − 3 y + z « 11 && Modulus « 5, 8x<D
H ζητάμε η εξίσωση να λυθεί ως προς το x L

88Modulus → 5, y → 3 + 4 x + 2 z<<
```

```
Solve@2 x^2 − 3 y + z « 11 && Modulus « 5, 8y<D
H ΜΗ γραμμική. Εδώ ζητάμε η εξίσωση να λυθεί ως προς y L

88Modulus → 5, y → 3 + 4 x^2 + 2 z<<
```

Αν χρησιμοποιούμε την έκδοση 5 του *Mathematica* τότε θα μπορούσαμε να αντικαταστήσουμε την Solve με την Reduce και να γράφαμε.

```
Reduce@2 x^2 - 3 y + z « 11, 8x, y, z<, Modulus -> 5D
```

Ειδικά σε περιπτώσεις της μορφής a*x-b=0(mod m) θα μπορούσαμε απλά να γράφαμε Solve[ax==b && Modulus==m, x] για να βρούμε το x. Άλλος τρόπος θα ήταν να πολλαπλασιάζαμε και τα δυο μέλη της a*x=b(mod m) με το d= a^{-1}(mod m) δηλ. με τον αντίστροφο του a . Στο *Mathematica* αυτό βρίσκεται με την συνάρτηση PowerMod δηλαδή d=PowerMod[a,-1,m]. Οπότε το ζητούμενο x=d*b(mod m). Για να λύσουμε για παράδειγμα την 5x=3(mod 7) γράφουμε

```
d = PowerMod@5, -1, 7D;H  εύρεση του 5^-1 H mod 7L  L
x = Mod@d 3, 7DH  x= d 3 Hmod 7L   L

2
```

Τελειώνουμε με την περίπτωση που έχουμε σύστημα γραμμικών εξισώσεων με διαφορετικά mod και ένα άγνωστο π.χ $x \equiv 0 \bmod 4$, $x \equiv 1 \bmod 9$, και $x \equiv 2 \bmod 121$. Τότε μπορούμε να χρησιμοποίσουμε την Chinese-Remainder. Π.χ γράφοντας ChineseRemainder[{0, 1, 2}, {4, 9, 121}] όπου η λίστα {0,1,2} είναι οι σταθερές και {4,9,121} τα modulus παίρνουμε την λύση x=244

```
<<NumberTheory`NumberTheoryFunctions`
ChineseRemainder[{0, 1, 2}, {4, 9, 121}]

244
```

Προσέξτε ότι χρειάστηκε να καλέσουμε πρώτα το πακέτο <<NumberTheory`NumberTheoryFunctions` πριν καλέσουμε την ChineseRemainder.

Κεφάλαιο 4ο: Γραμμική Άλγεβρα

Όpwν xέroum¶ énaν pínakaν ¶isάg¶tai ¶ít¶ m¶ ta stoic¶ía tou ¶ít¶ m¶ cr¶sh thν Table ¶ít¶ m¶ cr¶sh thν Array.

```
a = 881, 3, 2<, 84, 0, -1<<
b = Table@i^j, 8i, 3<, 8j, -1, 2<D
c = Array@#1^#2 &, 83, 4<D

881, 3, 2<, 84, 0, -1<<
```

```
981, 1, 1, 1<, 9i 1 j, 1, 2, 4=, 9i 1 j, 1, 3, 9==
                k 2 y                k 3 y
```

```
881, 1, 1, 1<, 82, 4, 8, 16<, 83, 9, 27, 81<<
```

To Array[#1^#2&,{3,4}] parάg¶i 3 gramméν m¶ 4 st¶l¶ν kai stoic¶ía $a_{ij} = i^j$. Gia na parάgoum¶ akribών to b qά pr¶p¶i na grάyoum¶

```
Clear@cD
c = Array@#1^#2 &, 83, 4<, 81, -1<D

981, 1, 1, 1<, 9i 1 j, 1, 2, 4=, 9i 1 j, 1, 3, 9==
                k 2 y                k 3 y
```

To {1,-1} sta dexiά shmaín¶i óti h prώth sunt¶tagménh x¶kinά¶i m¶ to 1 kai h d¶ót¶rh m¶ to -1. P.c

```
Clear@cD
c = Array@d, 83, 4<, 81, -1<D

88d@1, -1D, d@1, 0D, d@1, 1D, d@1, 2D<,
 8d@2, -1D, d@2, 0D, d@2, 1D, d@2, 2D<,
 8d@3, -1D, d@3, 0D, d@3, 1D, d@3, 2D<<
```

Fusikά, antí thν Array mporoúm¶ na crhsimopoiήsoum¶ thn Table s¶ kάq¶ p¶ríptwsh.

4.1 Baqmída dianusmátwn kai baqmída (táxh) pínaka

Baqmída twn dianusmátwn $v_1, v_2, ..., v_n$ **onomázoume thn diástash tou grammikoú cώrou pou parágetai apo touV grammikoúV sunduasmoúV twn dianusmátwn autών. Gia na broúm¶ thn baqmída kápoiwn dianusmátwn ¶kt¶loúm¶ stoic¶iώd¶iV grammoprάx¶iV ston pínaka me grammέV ta dianύsmata autά(metάqesh, prόsqesh ήajaíresh kάpoiaV grammήV se mia άllh k.o.k) étsi ώst¶ oi t¶l¶utaí¶V grammέV na gínoun mhd¶nikέV kai kάqe mh mhdenikήgrammήna xekinάei me monάda- thn monάda odhgó. Oi grammoprάx¶iV ¶kt¶loúntai m¶ thn sunάrthsh** RowReduce. **To plήqoV twn mh mhdenikών grammών pou prokύptoun eínai h zhtoúmenh baqmída**.

```
RowReduce@aD
RowReduce@bD

991, 0, -1/4=, 90, 1, 3/4==
```

```
881, 0, 0, 6<, 80, 1, 0, -11<, 80, 0, 1, 6<<
```

Den emjanízontai mhdenikέV grammέV. H baqmída touV eínai ísh me 3. **Ára kai stiV dύo p¶riptώs¶iV écoum¶ an¶xάrthta dianύsmata stiV grammέV twn** a,b. **An prosqέsoum¶ sthn** a **to diάnusma** {2,6,4} **tót¶ cán¶tai h gram. an¶xarthsía**:

```
d = Append@a, 82, 6, 4<D
d êê MatrixForm
RowReduce@dD

881, 3, 2<, 84, 0, -1<, 82, 6, 4<<
```

$$\begin{pmatrix} 1 & 3 & 2 \\ 4 & 0 & -1 \\ 2 & 6 & 4 \end{pmatrix}$$

```
991, 0, -1/4=, 90, 1, 3/4=, 80, 0, 0<<=
```

H mhdenikήgrammήdeícnei thn grmmikήexárthsh twn grammών tou pínaka d (**h tríth grammήeínai** 2 **jorέV thn** 1**h**)

H táxh enόV pínaka d **¶ínai ísh m¶ to plήqoV twn mh mhd¶nikών grammών tou** RowReduce[d]. **Sthn p¶ríptwsh maV loipón ¶ínai ísh m¶** 2. **Ára móno duo ap'autέV tiV gramm¶V ¶ínai gr. an¶xártht¶V**.

Parád¶igma: **Dín¶tai énaV** 5X5 **pínakaV** a **m¶ grammέV**

```
x = 81, 2, -1, 0, 1<; y = 82, 1, 0, 1, 3<; z = 80, 3, -2, -1, -1<;
t = 82, 4, -2, 0, 2<; s = 84, 5, -2, 1, 5<;
```

Diapistώs¶t¶ óti oi grammέV ¶ínai gr. **¶xarthmén¶V kai sthn sunέc¶ia na br¶q¶í énaV mh mhdenikόV grammikόV sunduasmόV touV pou na maV dín¶i to mhd¶nikó diάnusma**.

Lύsh: **Qa crhsimopoiήsoum¶ thn** RowReduce **gia na doúm¶ óti ¶ínai gr**. **¶xarthména kai sthn sunέc¶ia thn** Reduce **ήthn** LinearSolve **gia na lύsoum¶ to sύsthma** a.w=={0,0,0,0}. H Reduce[**¶xisώs,m¶tabl**] **aplopoi¶í tiV ¶xisώs(oi ¶xisώs mpor¶í na p¶rilambάnoun kai anisώs¶iV) wV proV tiV m¶tabl**. **Oi ¶xisώs¶iV pou prokύptoun ¶ínai isodύnam¶V m¶ tiV arcikέV**. **H** Reduce[**¶xisώs,m¶tabl,p¶díο**] **p¶riorίz¶i thn aplopoίhsh sto p¶díο(p.c p¶díο**=Integers)

```
Clear@aD; a = 8x, y, z, t, s<; RowReduce@aD

991, 0, 1/3, 2/3, 5/3=, 90, 1, -2/3, -1/3, -1/3=,
 80, 0, 0, 0, 0<, 80, 0, 0, 0, 0<, 80, 0, 0, 0, 0<=
```

Άra **¶ínai gr**. **¶xarthméneV kai h tάxh tou pίnaka** a **¶ínai ίsh m¶** 2

```
Reduce@w0 x + w1 y + w2 z + w3 t + w4 s « 0, 8w0, w1, w2, w3, w4<D

w0 == -2 Hw2 + w3 + w4L && w1 == w2 - w4
```

Όmoia **m¶ thn crήsh thV** LinearSolve

```
LinearSolve@a, 80, 0, 0, 0, 0<D

80, 0, 0, 0, 0<
```

Parathroúm¶ thn diajorά. **H** LinearSolve **maV édws¶ móno mία lύsh**, **thn mhdenikή**!

4.2 Grammikά sustήmata

Έstw óti **écoum¶ éna grammikó sύsthma thV morjήV** A.**C**=B ópou **A ¶ínai énaV** mXn **pínakaV kai B ¶ínai énaV** mX1 **pínakaV**. **To** m **eίnai to plήqoV twn exisώsewn kai to** n **twn agnώstwn**. **Gia na écI lύsh qa prέp¶i h tάxh tou A na ¶ínai ίsh m¶ thn tάxh tou ¶pauxhménou pínaka** (A|B).
p.c gia to sύsthma -2x+y+z=1, x-2y+z=-2, x+y-2z=4 **écoum¶**:

```
A = 88-2, 1, 1<,
    81, -2, 1<,
  81, 1, -2<<; B = 81, -2, 4<;
epayx = 88-2, 1, 1, 1<,  81, -2, 1, -2<, 81, 1, -2, 4<<
RowReduce@AD
RowReduce@epayxD
```

```
88-2, 1, 1, 1<, 81, -2, 1, -2<, 81, 1, -2, 4<<
```

```
881, 0, -1<, 80, 1, -1<, 80, 0, 0<<
```

```
881, 0, -1, 0<, 80, 1, -1, 0<, 80, 0, 0, 1<<
```

Parathroúm¶ óti oi dúo pínak¶V d¶n écoun thn ídia baqmída(táxh) ára to sústhma ¶ínai adúnaton.

Autó mproroúm¶ na ton diapistώsoum¶ kai m¶ állo trópo: **Zhtώntas m¶ thn** LinearSolve **na lúsei to sústhma**:

```
LinearSolve@A, BD
```

```
- LinearSolve::nosol : Linear equation encountered which has no solution.
```

```
LinearSolve@88-2, 1, 1<, 81, -2, 1<, 81, 1, -2<<, 81, -2, 4<D
```

Eidiká sthn p¶ríptwsh pou o A ¶ínai énas t¶tragwnikós pínakas tót¶ upárc¶i h p¶ríptwsh mias kai monadikήs lúshs. **Autó qa sumb¶í ótan o A kai o ¶pauxhménos écoun táxh akribώs ish m¶ thn diástash tou A dhl**. **m¶ to plήqos grammών tou**. **Bébaia gia t¶tragwnikoús A upárc¶i kai to kritήrio this ourízousas**: **An h** det[A] **¶ínai mh mhd¶nikή tót¶ káq¶ grammikó sústhma A.C=B écei ópws xéroum¶ mia monadikή lúsh thn C = A^-1 B**. **P.c o pínakas** $\begin{pmatrix} 2 & -1 & 3 \\ 1 & 3 & -2 \\ -1 & 11 & -12 \end{pmatrix}$ **d¶n ¶ínai antistréyimos**:

```
Clear@AD
A = 88 2  -1   3 <,
     81   3  -2 <,
     8-1  11 -12<<; Det@AD
```

```
0
```

opót¶ éna opoiodήpot¶ sústhma m¶ pínaka sunt¶l¶stών ton A p.c $A.X=\begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$ **d¶n écei monadikή lúsh**:

```
Clear@x, y, zD
Reduce@A.8x, y, z< « 81, 2, 4<, 8x, y, z<D

x == 1/7 H5 - 7 zL && y == 1/7 H3 + 7 zL
```

```
Solve@A.8x, y, z< « 81, 2, 4<, 8x, y, z<D

- Solve::svars : Equations may not give solutions for all "solve" variables.

99x → 5/7 - z, y → 3/7 + z==
```

H Solve **kai h** Reduce **einai scetikéV**. H Reduce **genikά upertereí diόti brískei όleV tiV dunatéV lύseiV**.

Άskhsh: **Na jtiacteí mia sunάrthsh** epayxhmenosMatrix[m_List,k_list] **όpou** m **kai** k **eínai dύo pínakeV kai pou qa epistréjei ton epauxhméno pínaka touV dhl**. **ton pínaka** m **ston opoío écoume episunάyei sta dexía twn sthlών tou**, **tiV stήleV tou** k. **Upόdeixh**. **Crhsimopoieíste katάllhla thn sunάrthsh** Append.

4.3 Oi IdiotimέV kai ta idioduanύsmata ¶nόV pínaka

Gia na broúm¶ ta idioduanύsmata ¶nόV t¶ragwnikoú pínaka A qa prέp¶i prώta na broúm¶ tiV idiotimέV tou dhl. **tiV ríz¶V tou carathristikoú poluwnόmou tou A**. **X¶kinάm¶ m¶ éna parád¶igma**. Έ**stw A**=$\begin{pmatrix} 3 & -2 & 0 \\ -2 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix}$**.Tót¶ to carakthristikó polυώnumo tou ¶ínai íso m¶** Det[A-x Identity-Matrix[3]] ό**pou to** IdentityMatrix[3]] **¶ínai o tautotikόV pínakaV** 3X3

```
Clear@AD
A = 88 3  -2  0<, 8-2  3  0<, 80  0  5<<
charPoly = Det@A - x IdentityMatrix@3DD
idiotimes = Solve@charPoly « 0, xD

883, -2, 0<, 8-2, 3, 0<, 80, 0, 5<<
```

```
25 - 35 x + 11 x^2 - x^3
```

```
88x → 1<, 8x → 5<, 8x → 5<<
```

M¶ ál la lógia écoum¶ dúo idiotimév thn r_1 = 5kai thn r_2 = 1pollaplóthtaV 2 **kai** 1 **antístoica**. Énav pio ¶úkoloV trópoV na brískoum¶ tiV idiotimév ¶ínai m¶ thn crήsh thV Eigenvalues:

```
Eigenvalues@AD

81, 5, 5<
```

Gia na broúm¶ mia lísta m¶ ta antístoica idioduanúsmata qa crhsimopoiήsoum¶ thn Eigenvectors:

```
Eigenvectors@AD

881, 1, 0<, 80, 0, 1<, 8-1, 1, 0<<
```

To próblhma m¶ thn Eigenvector ¶ínai óti d¶n mporoúm¶ na ¶ntopísoum¶ apo thn apánthsh poiá idioduanúsmata antistoicoún s¶ mia idiotimή. **Gia autó to lógo qa crhsimopoiήsoum¶ thn** NullSpace. **H** NullSpace[m] **maV dín¶i thn bάsh tou cώrou twn lús¶wn tou omog¶noúV sustήmatoV** m.X=**0**. **Opót¶** m¶ NullSpace[A-**l**IdentityMatrix[n]] (**όpou** n **h diάstash tou A**) **paírnoum¶ mia bάsh gia ta idionúsmata pou antistoicoún sthn idiotimή l p.c**.

```
bashldioxwroy@5D = NullSpace@A - 5 IdentityMatrix@3DD
bashldioxwroy@1D = NullSpace@A - 1 IdentityMatrix@3DD

880, 0, 1<, 8-1, 1, 0<<
```

```
881, 1, 0<<
```

Dhl. **mia bάsh tou idocώrou** (**tou cώrou twn idiodianusmάtwn**)**pou antistoic¶í sthn l**=5 **¶ínai h** {{0, 0, 1}, {-1, 1, 0}} **kai mia bάsh tou idocώrou pou antistoic¶í sthn l**=1 **¶ínai h** {{1,1,0}}. **T¶l¶iώnoum¶ thn ¶nóthta m¶ to anajéroum¶** óti **m¶ thn sunάrthsh** CharacteristcPolynomial **mporoúm¶ cwrίV kópo na broúm¶ to carakthristikó poluώnumo**:

```
Clear@tD
CharacteristicPolynomial@A, tD

25 - 35 t + 11 t^2 - t^3
```

4.4 H diagwnopoíhsh ¶nóV t¶tragwnikoú pínaka

H diagwnopoíhsh ¶nóV pínaka A éc¶i scésh m¶ tiV idiotimésV tou kai m¶ touV antístoicouV idiócwrouV. Eínai gnwstó apo thn q¶wría óti o A diagwnopoi¶ítai (**dhl**. **upárcei énaV antistréyimoV pínakaV** P **kai énaV diagónioV** D **étsi ώste** D=Invrse[P].A.P) **ann h pollaplóthta thV opoiasdήpote idiotimήV I tou A sumpípt¶i m¶ thn diástash tou antístoicou idiocώrou thV**. **Án káti tétoio iscú¶i tót¶** o **A diagwnopoi¶ítai kai o** D **éc¶i sthn diagónio tiV idiotimésV kai o** P **écei stiV stήleV tou ta antístoica idioduanósmata p.c**

```
P1 = Eigenvectors@AD
P = Transpose@P1D
Inverse@PD
diagwnios = Inverse@PD.A.P êê MatrixForm
```

```
881, 1, 0<, 80, 0, 1<, 8-1, 1, 0<<
```

```
881, 0, -1<, 81, 0, 1<, 80, 1, 0<<
```

```
99 1/2 , 1/2 , 0=, 80, 0, 1<, 9- 1/2 , 1/2 , 0==
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

M¶ thn ¶ukairía na anajéroum¶ óti h DiagonalMatrix[d]**dín¶i éna diagónio pínaka m¶ diagónio** d. **p.c**

```
DiagonalMatrix@Eigenvalues@ADD êê MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

O antistréyimoV pínakaV P **me thn idiόthta** P.diagwnios.Inverse[P]=A **den upárcei pánta** για κάθε πίνακα A. **Autó pou eínai gnwstó apo thn Grammikή'Algebra eínai óti upárcoun dύo pínakeV** R **kai** Q **kai énaV diagónioV** D =Diagonal[m] **étsi ώste**:
A=Transpose[R].D.Q. **Oi pínakeV autoí mporoúme na touV broúme me thn sunárthsh** SingularValues[A]. H SingularValues[A] **epistréjei éna pínaka me stoiceía** {R,m,Q}. **Gia na crhsimopoiήsoume thn** SingularValues[A] **prépei ta stoiceía tou A na dínontai me upodiastolή! Parádeigma**:

```
N@AD
b = SingularValues@N@ADD

883., -2., 0.<, 8-2., 3., 0.<, 80., 0., 5.<<
```

```
888-0.707107, 0.707107, 0.<,
  80., 0., 1.<, 8-0.707107, -0.707107, 0.<<,
 85., 5., 1.<, 88-0.707107, 0.707107, 0.<,
  80., 0., 1.<, 8-0.707107, -0.707107, 0.<<<
```

```
N@AD « Transpose@b@@1DDD.DiagonalMatrix@b@@2DDD.b@@3DD
b@@1DD êê MatrixForm
DiagonalMatrix@b@@2DDD êê MatrixForm
b@@3DD êê MatrixForm

True
```

```
-0.707107   0.707107   0.
    0.          0.     1.
-0.707107  -0.707107   0.
```

```
5.   0   0
 0   5.  0
 0   0   1.
```

```
-0.707107   0.707107   0.
    0.          0.     1.
-0.707107  -0.707107   0.
```

Parathroúme óti sthn períptwsh pou énaV pínakaV A diagwnopoieítai tóte o Transpose[R] **kai o** Q **pou dínei h** SingularValues **sumpíptoun**!

4.5 Εύρεση δυνάμεων πινάκων

Η διαγωνοποίηση είναι χρήσιμη για την γρήγορη εύρεση δυνάμεων τετραγωνικών πινάκων. Για παράδειγμα στο παραπάνω παράδειγμα υψώνοντας τις ιδιοτιμές στην διαγώνιο εις την 10η μπορούμε να βρούμε την 10η δύναμα του A: A^{10} = P.DiagonalMatrix@8$1^{10}$, 5^{10}, 5^{10}<D.Inverse@PD

```
P.DiagonalMatrix@81^10, 5^10, 5^10<D.Inverse@PD êê MatrixForm
```

$$\begin{pmatrix} 4882813 & -4882812 & 0 \\ -4882812 & 4882813 & 0 \\ 0 & 0 & 9765625 \end{pmatrix}$$

Με την ευκαιρία να αναφέρουμε ότι με `A^10`παίρνουμε

```
A^10
```

```
88 59049, 1024, 0<, 81024, 59049, 0<, 80, 0, 9765625<<
```

δηλ. αποτέλεσμα διαφορετικό απο αυτό που βρήκαμε πριν. Αυτό δεν σημαίνει ότι έχουμε κάνει λάθος παραπάνω. Οφείλεται στο γεγονός ότι στο *Mathematica* ο πολλαπλασιασμός A*A δεν είναι ο γνωστός μας πολλαπλασιασμός πινάκων. Ο γνωστός μας πολλαπλασιαμός πινάκων γίνεται με το Dot[A,B] που συμβολίζεται απλά με A.B Γενικά την n-ιοστή δύναμη του πίνακα A μπορούμε να την ορισουμε αναδρομικά ή αλλιώς μπορούμε να χρησιμοποιήσουμε την συνάρτηση Nest π.χ

```
pollaplasiasmos@a_D := A.a
Nest@pollaplasiasmos, A, 9D

88 4882813, -4882812, 0<, 8-4882812, 4882813, 0<, 80, 0, 9765625<<
```

```
?Nest
```

```
Nest@f, expr, nD gives an
  expression with f applied n times to expr.
```

Δηλαδή η Nest επιστρέφει το f(f(...f(expr))...))) όπου το f έχει εφαρμοστεί n φορές στην expr. Στην προηγούμενη χρήση του Nest εφαρμόσαμε 9 φορές το pollaplasiasmos διότι ήδη μέσα στο pollaplasiasmos υπάρχει ήδη 1 εφαρμογή του πολλαπλασιασμού με τον A.

4.6 Allagή thV bάshV tou $\tilde{N}^n$

Έstw óti maV dínoun duo bás¶iV tou $\tilde{N}^n$. Tót¶ to pιrasma apo thn mía bάsh sthn άllh p¶rigrάj¶tai m¶ ιna antistrιyimo pínaka P pou lιg¶tai pínakaV m¶tάbashV.AV doúm¶ ιna parάd¶igma. Dín¶tai mia bάsh B_1 tou $\tilde{N}^4$ kai o pínakaV metάbashV apo thn sunήqh bάsh sthn B_1 eínai o $P=\begin{pmatrix} 1 & 1 & 0 & -1 \\ -1 & 2 & 1 & 0 \\ 2 & -1 & 1 & -2 \\ -2 & -2 & 0 & 3 \end{pmatrix}$.

Na breqoún oi suntetagmιneV tou dianύsmatoV {1,2,3,4} **sthn paliά(sunήqh) bάsh. Apάnthsh: Oi suntetagmιneV eínai to ginόmeno** P.{1,2,3,4}. **ÁV doúme tiV práxeiV**

```
P = {{ 1   1   0  -1 }
     {-1   2   1   0 }
     { 2  -1   1  -2 }
     {-2  -2   0   3 }}
```

```
{{1, 1, 0, -1}, {-1, 2, 1, 0}, {2, -1, 1, -2}, {-2, -2, 0, 3}}
```

```
P.{1, 2, 3, 4}
```

```
{-1, 6, -5, 6}
```

Oi suntetagménεV tou $\{1,0,0,0\}$ **(dhl. tou prώtou basikoύ dianύsmatoV thV** B_1 **Lsth sunήqh bάsh einai**:

```
P.{1, 0, 0, 0}
```

```
{1, -1, 2, -2}
```

Autή eínai h prώth stήlh tou P. **Όmoia diapistώnoume óti oi stήleV tou** P **eínai oi suntetagménεV twn dianusmάtwn thV néaV bάshV wV proV thn sunήqh** βάsh. **AVdoύme to antrístrojo próblhma**: **Dínetai éna diάnusma me suntetagménεV** $\{-1, 6, -5, 6\}$ **wV proV thn sunήqh bάsh. PoiέV eínai oi suntetagménεV tou sthn néa bάsh**;

Skejtόmastε wV exήV epeidή $\{-1, 6, -5, 6\} = P.\{x_1, x_2, x_3, x_4\}$ **qa prέpei** $P^{-1}.\{-1, 6, -5, 6\} = \{x_1, x_2, x_3, x_4\}$.

```
Inverse@PD.{-1, 6, -5, 6}
```

```
{1, 2, 3, 4}
```

dhl. autó pou periméname. Genikά qa prépei o pínakaV metábashV P apó mia bάsh se mia άllh na eínai énaV antistréyimoV pínakaV. AV doúme άllo éna parάdeigma: Dínetai h bάsh B_2 pou oi suntetagméneV twn basikών dianusmάtwn eínai oi stήleV tou pínaka Q =

$$\begin{pmatrix} -2 & 0 & -2 & 1 \\ -1 & 1 & 0 & -2 \\ 1 & 2 & -1 & -1 \\ -2 & 2 & 1 & -2 \end{pmatrix}$$

. Ajoú deíxete óti prάgmati apoteloún bάsh na breíte

ton pínaka metάbashV apo thn B_1 sth B_2.

Apάnthsh: Kat' arcήn qa elégxoume thn orízousa tou Q. An eínai mh mhdenikή tóte oi stήleV eínai grammikώV anexάrhteV kai άra apoteloún mia bάsh. Gia na broúme ton pínaka metάbashV apo thn B_1 sth B_2 brískoume prώta ton pínaka metάbashV apo thn B_1 sthn sunήqh HautóV eínai o Inverse@PDL kai apo thn sunήqh sthn B_2 Hmésw tou QL. Telikά o zhtoúmenoV eínai o Inverse@PD.Q

```
Q = {{-2, 0, -2, 1}, {-1, 1, 0, -2}, {1, 2, -1, -1}, {-2, 2, 1, -2}}
```

```
{{-2, 0, -2, 1}, {-1, 1, 0, -2}, {1, 2, -1, -1}, {-2, 2, 1, -2}}
```

```
Det@QD
```

```
25
```

```
Inverse@PD
```

```
{{13/6, -1/6, 1/6, 5/6}, {5/6, 1/6, -1/6, 1/6},
 {1/2, 1/2, 1/2, 1/2}, {2, 0, 0, 1}}
```

```
MatrixForm@Inverse@PD.QD
```

$$\begin{pmatrix} -\frac{17}{3} & \frac{11}{6} & -\frac{11}{3} & \frac{2}{3} \\ -\frac{7}{3} & \frac{1}{6} & -\frac{4}{3} & \frac{1}{3} \\ -2 & \frac{5}{2} & -1 & -2 \\ -6 & 2 & -3 & 0 \end{pmatrix}$$

Genikά mporoúme na jtiάxoume mia sunάrthsh me eísodo duo doqeíseV báseiV V_1, V_2(ήna to poúme kalύtera me eísodo tiV suntetagméneV twn basikών dianusmάtwn twn V_1, V_2 ώV proV thn sunήqh bάsh)kai qa maV epistréjei ton pínaka metάbashV apo thn mia sthn άllh.

```
changeBasis@V1_List, V2_ListD :=
 If@Det@V1D != 0, Inverse@Transpose@V1DD.Transpose@V2D,
  "η αλλαγήβάσεως δεν είναι δυνατή"D
```

```
                  ¥ 1  -1  2  -2 ¥   ¥ -2  -1  1  -2 ¥
                  ¥ 1   2  -1 -2 ¥   ¥  0   1  2   2 ¥
changeBasisA¥ 0   1  1   0 ¥, ¥ -2  0  -1  1 ¥E
                  ¥-1   0  -2  3 ¥   ¥  1  -2  -1 -2 ¥

99- 17 , 11 , - 11 , 2 =, 9- 7 , 1 , - 4 , 1 =,
    3   6    3    3      3   6   3   3
 9-2, 5 , -1, -2=, 8-6, 2, -3, 0<=
      2
```

Άskhsh: O prosarthménoV pínakaV(adjoint) **enóV tetragwnikoú pínaka A sumbolízetai me** adj[A] **kai écei stoiceía ta** $b_{i,j} = (-1)^{i+j} D_{j,i}$ **ópou me** $D_{j,i}$ **sumbolízoume thn orízousa tou A ótan apo ton A ajaireqeí h** j **grammήkai h** i **stήlh. Kataskeuásete mia sunárthsh** minorMatrix[m_List,i_Integer/;-Positive[i],j_Integer/;Positive[j]] **pou me eísodo ta** m, i,j **qa epistréjei ton ellásona pínaka tou** m **cwríV thn** i **grammήkai thn** j **stήlh. Sthn sunéceia kataskeuásete thn sunárthsh** adjointMatrix[m] **pou qa epistréjei ton prosarthméno pínaka tou** m **kai dokimáste(gia éna sugkekriméno** m)**an iscúei h isóthta**:
m.adjointMatrix[m] ==Det[m].IdentityMatrix[Length[m]]==adjointMatrix[m].m **Upódeixh**: **Gia thn kataskeuήthV** minorMatrix **mporeíte na crhsimopoiήsete tiV sunartήseiV** Drop **kai** Part **pou eídame se prohgoúmeno máqhma**.

4.7 GrammikέV sunartήseiV kai pínakeV

Se autήn thn enóthta qa meletήsoume touV pínakeV apo mia állh skopiά. Káqe pínakaV A diastásewn mXn **orízei mia grammikήsunárthsh** $f : \mathbb{R}^n \to \mathbb{R}^m$ **me orismó** $f[e] = A.\{x_1, .. x_n\}$. (**me** $\{x_1, .. x_n\}$ **ennoούme tiV suntetagméneV tou dianύsmatoV** e **tou** $\mathbb{R}^n$ **wV proV thn sunήqh bάsh tou** $\mathbb{R}^n$. **Akóma me** f[e] **den ennooúme kápoio diánusma** e_1 **tou** $\mathbb{R}^m$ **allά tiV suntetagméneV tou** e_1 **wV proV thn sunήqh bάsh tou** $\mathbb{R}^m$). **Antístroja káqe grammikή** $f : \mathbb{R}^n \to \mathbb{R}^m$ **antistoicεί se éna pínaka A. AV doúme éna parádeigma**:

```
    ¥ 2   3  -1 ¥
A = ¥ 3  -1   2 ¥
    ¥ 1   2   3 ¥

882, 3, -1<, 83, -1, 2<, 81, 2, 3<<
```

tóte h sunárthsh matrixToFunction **pou orízetai parakátw ton metatrépei se grammikήsunárthsh**:

```
matrixToFunction@m_ListD := m.Table@xi, 8i, Length@First@mDD<D
f = matrixToFunction@AD

82 x1 + 3 x2 - x3, 3 x1 - x2 + 2 x3, x1 + 2 x2 + 3 x3<
```

To Length[First[m]] **eínai íso me to plήqoV twn sthlών tou** m. **Gia to antístrojo próblhma prépei na jtiάxoume mia sunάrthsh** functionToMatrix **pou na maV epistréjei ton pínaka pou krύbetai pίsw apo mia grammikήsunάrthsh** f. **Gia ton skopό autό qa creiastoύme thn** Variables[f] **pou epistréjei tiV metablhtέV thV** f **kai thn** Coefficient[g,**lίsta**] **pou dίnei touV suntelestέV twn metablhtών(ήtwn dunάmewn metablhtών) thV lίsta sthn sunάrthsh** g. **P.c**

```
Coefficient@2 x + 5 y^2, 8x<D
Coefficient@2 x + 5 y^2, 8x, y<D
Coefficient@2 x + 5 y y, 8x, y^2<D

82<
```

```
82, 0<
```

```
82, 5<
```

Parathreίste ό**ti sto poluώnumo** 2 x +5 y^2 **o** Coefficient **tou** y **eίnai** 0(**kai** ό**ci** 5 y) **enώ tou** y^2**eίnai** 5! H functionToMatrix **orίzetai wV ex'hV**:

```
functionToMatrix@f_ListD :=
 Table@Coefficient@f@@iDD, Variables@fDD, 8i, Length@fD<D
functionToMatrix@fD

882, 3, -1<, 83, -1, 2<, 81, 2, 3<<
```

Άskhsh: **Dίnetai h grammikήsunάrthsh** f(e)={x+2y, y-x,2x} ό**pou** x, y, z **eίnai oi suntegméneV tou** e **wV proV thn sunήqh bάsh**. **Breίte ton tύpo thV** f ό**tan allάxoume thn sunήqh bάsh tou** Ñ^2**sthn bάsh** B **=** 8v_1, v_2< ό**pou** v_1={1,1} **kai** v_2 **=** 81, 2< **kai thn bάsh tou** Ñ^3(**tou pedίou timών thV** f) **sthn B* =** 8u_1, u_2, u_3<ό**pou** u_1 **=** 81, 1, 1<, u_2 **=** 81, 1, 0<, u_3 **=** 81, 0, 0<.

Upόdeixh: Έ**stw ena tucaίo diάnusma** q **grammέno wV proV thn bάsh B**. **Oi suntetagméneV tou wV proV thn sunήqh bάsh eίnai** e^* **=**8x^*, y^*, z^*<=Transpose[B].q Ό**pόte me** fHe^*L **briskome tiV suntetagméneV thV eikόnoV(wV proV thn sunήqh bάsh) kai me** Inverse@Transpose@B^*DD.fHe^*L **briskoume tiV zhtoύmeneV suntetagméneV wV proV thn bάsh B***.

Άllo**V trόpoV**: έ**stw A o pίnakaV thV** f **wV proV tiV sunήqeiV bάseiV kai** έ**stw** P = transpose[B]**o pίnakaV metάbashV apo thn sunήqh bάsh tou** Ñ^2**sthn** B **kai** έ**stw** Q **=** Transpose@B^*Do **pίnakaV metάbashV apo thn sunήqh bάsh tou** Ñ^3**sthn B***. **Tόte eίnai gnwstό apo thn qewrία ό**t**i o pίnakaV thV** f **wV proV tiV bάseiV B sthn B*eίnai** ίsoV **me A* =** Q^{-1}.A.P. **Apo edώ mporoύme eύkola na broύme ton néo tύpo thV** f **me éna aplό pollaplasiasmό** f^*HqL **=** A^*.q

Κεφάλαιο 5ο: Επίλυση εξισώσεων και συστημάτων

5.1 Επίλυση εξισώσεων

Το *Mathematica* διαθέτει αρκετές συναρτήσεις για την επίλυση εξισώσεων. Αυτές είναι:

Solve[eqn, x]	επιλύνει την εξίσωση eqn με άγνωστο το x.
Reduce[eqn, x]	επιλύνει την εξίσωση eqn με άγνωστο το x και κάνει διερεύνηση.
NSolve[eqn, x]	επιστρέφει αριθμητικές λύσεις της εξίσωσης eqn.
FindRoot[eqn, {x, x_0}]	επιστρέφει μια προσεγγιστική λύση της εξίσωσης eqn στην περιοχή του σημείου x_0.
Roots[eqn, x]	επιλύνει την πολυωνιμική εξίσωση eqn με άγνωστο το x.

Με τις συναρτήσεις **Solve**, **Reduce** και **Roots** βρίσκουμε ακριβείς λύσεις των εξισώσεων ενώ με τις συναρτήσεις **NSolve** και **FindRoot** βρίσκουμε μόνο αριθμητικές (προσεγγιστικές) λυσεις.

Με ακρίβεια μπορούμε να επιλύσουμε ένα μεγάλο πλήθος διαφορετικών εξισώσεων, αλλά όχι όλα τα είδη. Π.χ. μια πολυωνυμική εξίσωση μέχρι τετάρτου βαθμού επιλύνεται πάντοτε με ακρίβεια, αλλά μια πολυωνυμική εξίσωση βαθμού ανωτέρου του τετάρτου μπορεί να λυθεί με ακρίβεια αλλά μπορεί και όχι.

Προσεγγιστικά μπορούμε να λύσουμε όλα τα είδη των εξισώσεων και συστημάτων.

5.1.1 Ακριβής επίλυση εξισώσεων

Η βασική συνάρτηση επίλυσης εξισώσεων με ακρίβεια είναι η **Solve**. Η εξίσωση στο *Mathematica* εισάγεται με διπλό σύμβολο ισότητας ==, αφού το απλό σύμβολο της ισότητας χρησιμοποιείται από πρόγραμμα για ορισμούς. Οι λύσεις x_0, x_1, ... εμφανίζονται σε λίστα υπό τη μορφή κανόνων αντικατάσταση "**Ø**", ως εξής:

$$\{\{x \text{Ø} x_0\}, \{x \text{Ø} x_1,\}, ...\}.$$

Έστω η εξίσωση τρίτου βαθμού $x^3 + 4\,x^2 - 11\,x - 30 = 0$, την οποία αποθηκεύουμε στην μεταβλητή eqn.

```
eqn = x^3 + 4 x^2 - 11 x - 30 « 0;
```

Επειδή στο τέλος της προηγούμενης εντολής υπάρχει το ";", δεν εμφανίζεται το output.

Στη συνέχεια επιλύουμε την εξίσωση eqn1 ως προς x (με τη χρήση της συνάρτησης **Solve**) και τη λύση την αποθηκεύουμε στην μεταβλητή sol.

```
sol = Solve@eqn, xD

88x → −5<, 8x → −2<, 8x → 3<<
```

Επαλήθευση μπορούμε να κάνουμε είτε με την εντολή

```
x^3 + 4 x^2 − 11 x − 30 ê. sol

80, 0, 0<
```

είτε με την εντολή.

```
eqn ê. sol

8True, True, True<
```

Επειδή συνήθως είναι πιο βολικό να έχουμε τις λύσεις x_0, x_1, ... σε μορφή λίστα $\{x_0, x_1, \ldots\}$, χωρίς κανόνες αντικατάστασης, μπορούμε να χρησιμοποιήσουμε τον τελεστή αντικατάστασης "/." για να το επιτύχουμε. Πολλές φορές, μάλιστα, είναι σκόπιμο να δώσουμε στη λίστα που θα προκύψει ένα όνομα.

```
X = x ê. sol

8−5, −2, 3<
```

Με τον τρόπο αυτό, δηλαδή έχοντας τις λύσεις στη διάθεση μας ως λίστα, μπορούμε να κάνουμε πράξεις:

```
X + 5

80, 3, 8<
```

```
X^2

825, 4, 9<
```

η να αναφερθούμε σε κάποια συγκεκριμένη λύση, γράφοντας κατά τα γνωστά X[[i]], i = 1, 2, 3 ... Π.χ. για να πάρουμε τη λύση x = -5 γράφουμε

```
X@@1DD

-5
```

Επίσης, με τη βοήθεια της λίστας, η επαλήθευση για κάποια συγκεκριμένη λύση (π.χ. για x = -2) γίνεται είτε με την εντολή:

```
x^3 + 4 x^2 - 11 x - 30 /. x -> X@@2DD

0
```

είτε με την εντολή:

```
eqn /. x → X@@2DD

True
```

Αν μια λύση είναι διπλή (ή τριπλή κ.λ.π.) η Solve την επιστρέφει δύο (ή τρεις κ.λ.π.) φορές στη λίστα των λύσεων, ενώ όταν δεν υπάρχουν λύσεις, επιστρέφει { }.

Τις ίδιες εξισώσεις,που επιλύνει η συνάρτηση **Solve**,επιλύνει και η συνάρτηση **Reduce**. Οι διαφορές ανάμεσα στην συνάρτηση **Reduce** και στην συνάρτηση **Solve** είναι οι εξής:

1. Η συνάρτηση **Reduce** παρουσιάζει τις λύσεις υπό τη μορφή:

$$x == x_0 \,||\, x == x_1 \ldots.$$

(θυμίζουμε ότι | | είναι ο λογικός τελεστής, που παριστάνει το διαζευτικό ή).

2. Όταν η εξίσωση περιλαμβάνει μια ή περισσότερες παραμέτρους, η **Reduce** την επιλύνει παρουσιάζοντας όλες τις δυνατές περιπτώσεις, κάνοντας δηλαδή ουσιαστικά διερεύνηση, κάτι που δεν κάνει η **Solve**.

Παράδειγμα 1: Να λυθεί η εξίσωση $x^3 - x^2 - 4 = 0$.

Επίλυση της εξίσωσης με τη συνάρτηση **Solve**:

```
Solve@x^3 - x^2 - 4 « 0, xD

98x → 2<, 9x → i1 I-1 - ą 7M=, 9x → i1 I-1 + ą 7M==
                 2                  2
```

Μετατροπή των λύσεων από κανόνες αντικταστάσης σε λίστα:

```
x ê. %

92, 1/2 I-1 - ą è!!! 7M, 1/2 I-1 + ą è!!! 7M=
```

Οι λύσεις που παίρνουμε με τη συνάρτηση Solve είναι ακριβείς. Φυσικά μπορούμε να πάρουμε αμέσως αριθμητικές (προσεγγιστικές) λύσεις με τη συνάρτηση **N**, και μάλιστα με όση ακρίβεια δεκαδικών επιθυμούμε:

```
N@%%D

88x → 2.<, 8x → -0.5 - 1.32288 ą<, 8x → -0.5 + 1.32288 ą<<
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce**:

```
Reduce@x^3 - x^2 - 4 « 0, xD

x == 2 »» x == 1/2 I-1 - ą è!!! 7M »» x == 1/2 I-1 + ą è!!! 7M
```

Για να πάρουμε τις λύσεις σε μορφή λίστας, εφαρμόζουμε πρώτα την εντολή {**ToRules**[%]}, για να τις εμφανίσουμε ως κανόνες αντικατάστασης, και στη συνέχεια εφαρμόζουμε τον τελεστή αντικταστάσης "/.":

```
8ToRules@%D<

98x → 2<, 9x → 1/2 I-1 - ą è!!! 7M=, 9x → 1/2 I-1 + ą è!!! 7M==
```

```
x ê. %

92, 1/2 I-1 - ą è!!! 7M, 1/2 I-1 + ą è!!! 7M=
```

Ισοδύναμα μπορούμε να εφαρμόσουμε και τον σύνθετο τελεστή x /. {**ToRules**[%]}:

```
x ê. 8ToRules@%%%D<

92, 1/2 I-1 - ą è!!! 7M, 1/2 I-1 + ą è!!! 7M=
```

Εύρεση αριθμητικών (προσεγγστικών) λύσεων:

```
N@%%%D

88x → 2.<, 8x → −0.5 − 1.32288 ą<, 8x → −0.5 + 1.32288 ą<<
```

Παράδειγμα 2: Να λυθεί η εξίσωση $a x^2 - 2 x + 4 = 0$.

Επίλυση της εξίσωσης με τη συνάρτηση **Solve**:

```
Solve@a x^2 − 2 x + 4 « 0, xD

99x → 2 − è!!!!!!!!!!!!!!!!4 − 16 a / 2 a =, 9x → 2 + è!!!!!!!!!!!!!!!!4 − 16 a / 2 a ==
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce**:

```
Reduce@a x^2 − 2 x + 4 « 0, xD

a == 0 && x == 2 »» x == 2 − è!!!!!!!!!!!!!!!!4 − 16 a / 2 a && a ≠ 0 »» x == 2 + è!!!!!!!!!!!!!!!!4 − 16 a / 2 a && a ≠ 0
```

Μετατροπή των λύσεων σε κανόνες αντικατάστασης:

```
8ToRules@%D<

98a → 0, x → 2<, 9x → 2 − è!!!!!!!!!!!!!!!!4 − 16 a / 2 a =, 9x → 2 + è!!!!!!!!!!!!!!!!4 − 16 a / 2 a ==
```

Βλέπουμε ότι επιλύοντας την εξίσωση με τη συνάρτηση **Solve**, το *Mathematica* τη θεωρεί δευτεροβάθμια, δηλαδή υποθέτει ότι a ∫ 0 ενώ επιλύοντας την εξίσωση με τη συνάρτηση **Reduce** γίνεται πλήρης διερεύνηση. Συγκεκριμένα, βρέθηκε επιπλέον η λύση x = 2, η οποία προκύπτει όταν a = 0 (δηλαδή όταν η εξίσωση είναι πρωτοβάθμια).

5.1.1.1 Πολυωνυμικές εξισώσεις

Με τις συναρτήσεις **Solve** και **Reduce** μπορούμε να επιλύσουμε πολλά διαφορετικά είδη εξισώσεων, αλλά τόσο αυτές οι δύο όσο και η συνάρτηση **Roots**, που θα γνωρίσουμε παρακάτω, είναι ιδιαίτερα κατάλληλες για την επίλυση πολυωνυμικών εξισώσεων.

Οι συναρτήσεις **Solve** και **Reduce** επιλύνουν με ακρίβεια κάθε πολυωνυμική εξίσωση, η οποία είναι βαθμού n **§** 4. Όταν, όμως, είναι n **¥** 5, τότε δεν τις επιλύνουν πάντα, χωρίς βέβαια να αποκλείεται κάτι τέτοιο. Στην περίπτωση αυτή το *Mathematica* χρησιμοποιεί εκφράσεις της μορφής Root για να τις αναπαραστήσει. Όταν συμβεί κάτι τέτοιο, μπορούμε να πάρουμε εκ των υστέρων αριθμητικές τιμές για τις λύσεις με τη συνάρτηση **N**.

Παράδειγμα 3: Να λυθεί η εξίσωση $x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$.

Επιλύνουμε πρώτα την εξίσωση με τη συνάρτηση **Solve**:

```
Solve@x^4 + 2 x^3 - 13 x^2 - 14 x + 24 « 0, xD

88x → -4<, 8x → -2<, 8x → 1<, 8x → 3<<
```

και στη συνέχεια με τη συνάρτηση **Reduce**:

```
Reduce@x^4 + 2 x^3 + -13 x^2 - 14 x + 24 « 0, xD

x == -4 »» x == -2 »» x == 1 »» x == 3
```

Μετατροπή των λύσεων σε κανόνες αντικατάστασης:

```
8ToRules@%D<

88x → -4<, 8x → -2<, 8x → 1<, 8x → 3<<
```

Παρατηρούμε, ότι και οι δύο επέστρεψαν τις ίδιες λύσεις.

Παράδειγμα 4: Να λυθεί η εξίσωση $x^5 + 4x - 1 = 0$.

Επιλύνουμε πρώτα την εξίσωση με τη συνάρτηση **Solve**:

```
Solve@x^5 + 4 x - 1 « 0, xD

88x → Root@-1 + 4 #1 + #1^5 &, 1D<,
 8x → Root@-1 + 4 #1 + #1^5 &, 2D<, 8x → Root@-1 + 4 #1 + #1^5 &, 3D<,
 8x → Root@-1 + 4 #1 + #1^5 &, 4D<, 8x → Root@-1 + 4 #1 + #1^5 &, 5D<<
```

και στη συνέχεια με τη συνάρτηση **Reduce**:

```
Reduce@x^5 + 4 x - 1 « 0, xD

Root@-1 + 4 #1 + #1^5 &, 1D == x »»
 Root@-1 + 4 #1 + #1^5 &, 2D == x »» Root@-1 + 4 #1 + #1^5 &, 3D == x »»
 Root@-1 + 4 #1 + #1^5 &, 4D == x »» Root@-1 + 4 #1 + #1^5 &, 5D == x
```

Παρατηρούμε ότι το *Mathematica* δεν μπορεί να βρει τις ακριβείς λύσεις της εξίσωσης ούτε με τη συνάρτηση **Solve** ούτε με τη συνάρτηση **Reduce**. Μπορούμε όμως να πάρουμε αριθμητικές (προσεγγιστικές) λύσεις της εξίσωσης με τη συνάρτηση **N**:

```
N@%%D

88x → 0.249757<, 8x → -1.05775 - 1.00384 ą<,
 8x → -1.05775 + 1.00384 ą<,
 8x → 0.932871 - 1.00627 ą<, 8x → 0.932871 + 1.00627 ą<<
```

Παράδειγμα 5: Να λυθεί η εξίσωση $x^7 - x^5 + 4x^3 - x^2 - 4x + 1 = 0$.

Επιλύνουμε την εξίσωση με τη συνάρτηση **Solve**:

```
Solve@x^7 - x^5 + 4 x^3 - x^2 - 4 x + 1 « 0, xD

88x → -1<, 8x → 1<, 8x → Root@-1 + 4 #1 + #1^5 &, 1D<,
 8x → Root@-1 + 4 #1 + #1^5 &, 2D<, 8x → Root@-1 + 4 #1 + #1^5 &, 3D<,
 8x → Root@-1 + 4 #1 + #1^5 &, 4D<, 8x → Root@-1 + 4 #1 + #1^5 &, 5D<<
```

Παρατηρούμε ότι το *Mathematica* επιστρέφει δύο ακριβείς λύσεις, τις $x_0 = -1$ και $x_1 = 1$ ενώ τις υπόλοιπες τις αναπαριστά με χρήση εκφράσεων της μορφής Root. Αυτό είναι αναμενόμενο, γιατί αν αναλύσουμε το πολυώνυμο σε γινόμενο πολυωνύμων μικρότερου βαθμού, με τη χρήση της συνάρτησης **Factor,** θα έχουμε το εξής αποτέλεσμα:

```
Factor@x^7 - x^5 + 4 x^3 - x^2 - 4 x + 1D

H-1 + xL H1 + xL H-1 + 4 x + x^5L
```

Εύρεση αριθμητικών (προσεγγιστικών) λύσεων της εξίσωσης:

```
N@%%D

88x → -1.<, 8x → 1.<, 8x → 0.249757<,
 8x → -1.05775 - 1.00384 ą<, 8x → -1.05775 + 1.00384 ą<,
 8x → 0.932871 - 1.00627 ą<, 8x → 0.932871 + 1.00627 ą<<
```

Εκτός από τις συναρτήσεις **Solve** και **Reduce**, μια άλλη συνάρτηση, που μπορούμε να χρησιμοποιήσουμε για να λύνουμε εξισώσεις, είναι η **Roots**, η σύνταξη της οποίας είναι παρόμοια με των άλλων δύο. Επιλύνει όμως **μόνον πολυωνυμικές εξισώσεις** και δίνει λύσεις της μορφής:

$$x == x_0 \,||\, x == x_1 \ldots.$$

Παράδειγμα 6: Να λυθεί η εξίσωση $a\,x^2 - 2\,x + 4 = 0$.

Επίλυση της εξίσωσης με τη συνάρτηση **Roots**:

```
Roots@a x^2 - 2 x + 4 « 0, xD

x == (2 - è!!(4 - 16 a)) / (2 a) »» x == (2 + è!!(4 - 16 a)) / (2 a)
```

Μετατροπή των λύσεων σε κανόνες αντικατάστασης:

```
8ToRules@%D<

99x → (2 - è!!(4 - 16 a)) / (2 a)=, 9x → (2 + è!!(4 - 16 a)) / (2 a)==
```

Παρατήρηση: Οι εντολές **Solve**[eqn,x] και {**ToRules**[**Roots**[eqn,x]]}, επιστρέφουν το ίδιο αποτέλεσμα.

```
Solve@a x^2 - 2 x + 4 « 0, xD

99x → (2 - è!!(4 - 16 a)) / (2 a)=, 9x → (2 + è!!(4 - 16 a)) / (2 a)==
```

```
8ToRules@Roots@a x^2 - 2 x + 4 « 0, xDD<

99x → (2 - è!!(4 - 16 a)) / (2 a)=, 9x → (2 + è!!(4 - 16 a)) / (2 a)==
```

Με τις συναρτήσεις **Solve** και **Reduce** μπορούμε να λύσουμε και εξισώσεις που περιλαμβάνουν είτε ριζικά είτε κλάσματα, όπως επίσης και τριγωνομετρικές εξισώσεις, εκθετικές και λογαριθμικές εξισώσεις.

5.1.1.2 Εξισώσεις με ριζικά

Παράδειγμα 7: Να λυθεί η εξίσωση $\frac{1}{x} - \frac{1}{e^x} = 2$.

Επίλυση της εξίσωσης με τη συνάρτηση **Solve**:

```
Solve[1/x - 1/E^x « 2, x]

{{x → 1/4}}
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce**:

```
Reduce[1/x - 1/E^x « 2, x]

x == 1/4 && x ≠ 0
```

Εύρεση αριθμητικών (προσεγγστικών) λύσεων της εξίσωσης:

```
N@%%D

{{x → 0.25}}
```

5.1.1.3 Κλασματικές εξισώσεις

Παράδειγμα 8: Να λυθεί η εξίσωση $\frac{x}{x^2-1} - \frac{1}{x+1} = \frac{x^2}{x^2+1}$.

Επίλυση της εξίσωσης με τη συνάρτηση **Solve**:

```
Solve[x/(x^2 - 1) - 1/(x + 1) == x^2/(x^2 + 1), x]

{{x → -Sqrt[-1 + Sqrt[2]]}, {x → Sqrt[-1 + Sqrt[2]]},
 {x → -Sqrt[1 + Sqrt[2]]}, {x → Sqrt[1 + Sqrt[2]]}}
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce**:

```
Reduce[x/(x^2 - 1) - 1/(x + 1) == x^2/(x^2 + 1), x]

x == -Sqrt[-1 + Sqrt[2]] || x == Sqrt[-1 + Sqrt[2]] || x == -Sqrt[1 + Sqrt[2]] || x == Sqrt[1 + Sqrt[2]]
```

Εύρεση αριθμητικών (προσεγγστικών) λύσεων της εξίσωσης:

```
N@%%D

88x → 0. - 0.643594 ą<,
 8x → 0. + 0.643594 ą<, 8x → -1.55377<, 8x → 1.55377<<
```

5.1.1.4 Τριγωνομετρικές εξισώσεις

Παράδειγμα 9: Να λυθεί η εξίσωση `Sin@xD = 1/2`.

Είναι γνωστό ότι η εξίσωση `Sin@xD = 1/2` έχει άπειρες λύσεις της μορφής x = π/6 + 2kπ, k œ Ñ. Αν επιχειρήσουμε να λύσουμε αυτή την εξίσωση με κάποια από τις συναρτήσεις **Solve** ή **Reduce**, θα πάρουμε αρχικά ένα μήνυμα, με το οποίο το πρόγραμμα μας προειδοποιεί ότι έχουν χρησιμοποιηθεί αντίστροφες συναρτήσεις κατά την επίλυση με συνέπεια κάποιες λύσεις μιθανόν να μην βρεθούν, και στη συνέχεια το πρόγραμμα παρουσιάζει μόνο μία λύση (αυτή που προκύπτει για k = 0).

Επίλυση της εξίσωσης με τη συνάρτηση **Solve**:

```
Solve[Sin@xD « 1/2, x]

– Solve::ifun :
   Inverse functions are being used by Solve, so some solutions may not be found.

99x → π/6 ==
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce**:

```
Reduce[Sin@xD « 1/2, x]

– Reduce::ifun :
   Inverse functions are being used by Reduce, so some solutions may not be found.

x == π/6
```

Οι συναρτήσεις **Solve** ή **Reduce**, έχουν μία επιλογή την **InverseFunctions**, η οποία καθορίζει κατά πόσο αντίστροφες συναρτήσεις μπορούν να χρησιμοποιηθούν κατά την επίλυση.

Οι ρυθμίσεις της επιλογής **InverseFunctions** είναι οι εξής:

True	αντίστροφες συναρτήσεις θα χρησιμοποιούνται πάντα.
Automatic	αντίστροφες συναρτήσεις θα χρησιμοποιούνται, αλλά θα εκτυπώνεται ένα μήνυμα (default).
False	αντίστροφες συναρτήσεις δεν θα χρησιμοποιούνται.

Όταν δεν χρησιμοποιούμε αυτή την επιλογή στις συναρτήσεις **Solve** ή **Reduce**, θεωρείται ότι η επιλογή **InverseFunctions** έχει τη ρύθμιση Automatic.

Επίλυση της εξίσωσης με τη συνάρτηση **Solve** και με χρήση της επλογής **InverseFunctios**:

```
Solve[ArcSin[x] == 1/2, x, InverseFunctions → True]

{{x → π/6}}
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce** και με χρήση της επλογής **InverseFunctios**:

```
Reduce[ArcSin[x] == 1/2, x, InverseFunctions → True]

x == π/6
```

Εύρεση αριθμητικής (προσεγγστικής) λύσης της εξίσωσης:

```
N[%%]

{{x → 0.523599}}
```

5.1.1.5 Εκθετικές και λογαριθμικές εξισώσεις

Παράδειγμα 10: Να λυθεί η εξίσωση $2 e^{x} + 4 e = 4 e^{x} + e 2^{x}$.

Επίλυση της εξίσωσης με τη συνάρτηση **Solve**:

```
Solve[2 e^x + 4 e == 4 e^x + e 2^x, x]

Solve::ifun :
 Inverse functions are being used by Solve, so some solutions may not be found.

{{x → 1}, {x → 2}}
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce** και με χρήση της επλογής **InverseFunctios**:

```
Reduce@H2 eL^x + 4 e « 4 e^x + e 2^x, x, InverseFunctions → TrueD

x == 1 »» x == 2
```

Παράδειγμα 11: Να λυθεί η εξίσωση $\ln\left(\sqrt{x}\right) = \sqrt{\ln(x)}$.

Επίλυση της εξίσωσης με τη συνάρτηση **Solve**:

```
Solve@Log@Sqrt@xDD « Sqrt@Log@xDD, xD

88x → 1<, 8x → Ą^4<<
```

Επίλυση της εξίσωσης με τη συνάρτηση **Reduce**:

```
Reduce@Log@Sqrt@xDD « Sqrt@Log@xDD, xD

x == 1 »» x == Ą^4
```

5.1.2 Αριθμητική (προσεγγιστική) επίλυση εξισώσεων

Δύο είναι οι βασικές συναρτήσεις με τις οποίες μπορούμε να πάρουμε αριθμητικές (προσεγγιστικές) λύσεις εξισώσεων. Η μία είναι η συνάρτηση **NSolve**[eqn, x], η οποία εφαρμόζεται κυρίως σε πολυωνυμικές και γενικά σε εξισώσεις, που μπορεί να επιλύσει η συνάρτηση **Solve**. Η άλλη είναι η συνάρτηση **FindRoot**[eqn, {x, x_0}], η οποία αναζητεί μια αριθμητική λύση της εξίσωσης στην περιοχή του x = x_0.

Παράδειγμα 12: Να λυθεί η εξίσωση $x^3 + 4x - 1 = 0$.

Για να βρούμε τις αριθμητικές λύσεις της εξίσωσης, μπορούμε να τη λύσουμε πρώτα με τη συνάρτηση **Solve**, για να πάρουμε τις ακριβείς λύσεις της (κάτι που είναι εφικτό αφού η εξίσωση είναι τρίτου βαθμού),

```
Solve@x^3 + 4 x − 1 « 0, xD

99x → −4 ikjj 2 / (3 I9 + è!!!!!!! 849 M) y{zz^1ê3 + I 1/2 I9 + è!!!!!!! 849 MM^1ê3 / 3^2ê3 =,
 9x → 2 I1 + ą è!!! 3 M ikjj 2 / (3 I9 + è!!!!!!! 849 M) y{zz^1ê3 − I1 − ą è!!! 3 M I 1/2 I9 + è!!!!!!! 849 MM^1ê3 / (2 3^2ê3) =,
 9x → 2 I1 − ą è!!! 3 M ikjj 2 / (3 I9 + è!!!!!!! 849 M) y{zz^1ê3 − I1 + ą è!!! 3 M I 1/2 I9 + è!!!!!!! 849 MM^1ê3 / (2 3^2ê3) ==
```

και μετά να μετατρέψουμε τις ακριβείς λύσεις σε αριθμητικές με τη συνάρτηση **N**:

```
N@%D

88x → 0.246266<, 8x → −0.123133 + 2.01134 ą<,
 8x → −0.123133 − 2.01134 ą<<
```

Αν δεν μας ενδιαφέρουν οι ακριβείς λύσεις, τότε εφαμόζουμε απ' ευθείας τη συνάρτηση **NSolve**:

```
NSolve@x^3 + 4 x − 1 « 0, xD

88x → −0.123133 − 2.01134 ą<,
 8x → −0.123133 + 2.01134 ą<, 8x → 0.246266<<
```

Με τη συνάρτηση **NSolve** μπορούμε να επιλύσουμε αριθμητικά σχεδόν όλα τα είδη εξισώσεων. Για παράδειγμα:

Παράδειγμα 13: Να λυθεί αριθμητικά η εξίσωση $2\,x^2 - \sqrt{x} = 2$.

```
NSolve@2 x^2 − Sqrt@xD « 2, xD

88x → 1.24847<<
```

Παράδειγμα 14: Να λυθεί αριθμητικά η εξίσωση $\frac{1}{x+1} + \frac{1}{x^2-1} = 2$.

```
          1      1
NSolveA¡¡¡¡¡¡¡¡¡¡ + ¡¡¡¡¡¡¡¡¡¡ « 2, xE
        x + 1    x^2 − 1

88x → 1.28078<, 8x → −0.780776<<
```

Παράδειγμα 15: Να λυθεί αριθμητικά η εξίσωση $\sin x + 2\cos x = 1$.

```
NSolve@Sin@xD + 2 Cos@xD « 1, xD
```

```
Solve::ifun :
 Inverse functions are being used by Solve, so some solutions may not be found.
```

```
88x → −0.643501<, 8x → 1.5708<<
```

Παράδειγμα 16: Να λυθεί αριθμητικά η εξίσωση $\ln\left(x^2 - \sqrt{1-x}\right) = 1$.

```
NSolve[Log[x^2 - Sqrt[1 - x]] == 1, x]

{{x → -2.11753}}
```

Η εντολή **NSolve** δεν βρίσκει αριθμητικές λύσεις για όλα τα είδη των εξισώσεων.

Παράδειγμα 17: Να λυθεί αριθμητικά η εξίσωση $\cos x - x^2 = 0$.

Παρατηρούμε ότι η εξίσωση δεν μπορεί να λυθεί αριθμητικά με τη χρήση της συνάρτησης **NSolve**.

```
NSolve@Cos@xD - x^2 == 0, xD

Solve::tdep : The equations appear to involve the
   variables to be solved for in an essentially non-algebraic way.

NSolve@-x^2 + Cos@xD == 0, xD
```

Σε τέτοιες περιπτώσεις, και όχι μόνο σε αυτές, χρησιμοποιούμε τη δεύτερη συνάρτηση που διαθέτει το *Mathematica*, τη συνάρτηση **FindRoot**[eqn, {x, x_0}], η οποία αναζητεί μια αριθμητική λύση της εξίσωσης στην περιοχή του x = x_0. Συγκεκριμένα, αναζητούμε μία αριθμητική λύση στη περιοχή του x = 1.

```
FindRoot@Cos@xD - x^2 == 0, 8x, 1<D

{x → 0.824132}
```

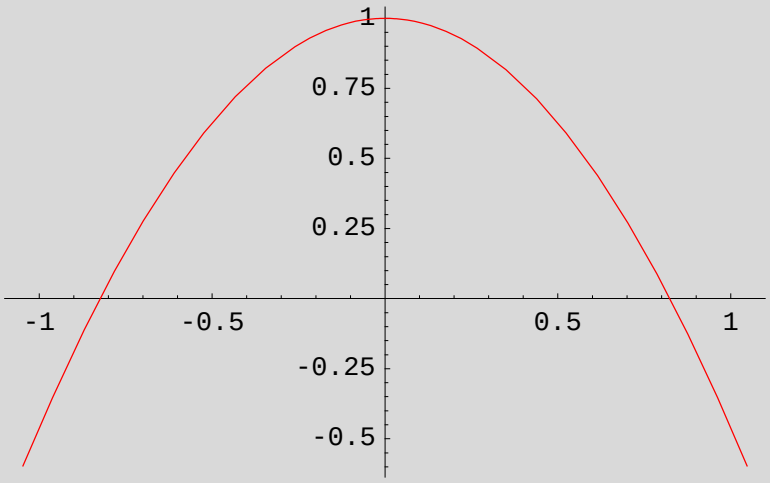
Μια καλή τακτική, που μπορούμε να ακολουθήσουμε σε συνδυασμό με τη συνάρτηση **FindRoot**, είναι να κάνουμε πρώτα τη γραφική παράσταση της συνάρτησης, της οποία ο μηδενισμός μας δίνει της δίνει την προ επίλυση εξίσωση, με τη συνάρτηση:

Plot[f(x), {x, xminx, xmax}]

(στην οποία θα αναφερθούμε αναλυτικά σε επόμενο κεφάλαιο), σε ένα κατάλληλο διάστημα [xmin, xmax]. Από τη γραφική παράσταση λαμβάνουμε κατά προσέγγιση τα σημεία τομής της με τον άξονα των x και τα χρησιμοποιούμε ως σημεία αφετηρίας στη συνάρτηση **FindRoot** (δηλαδή, αναζητούμε λύσεις στην περιοχή αυτών των σημείων).

Συγκεκριμένα, στο παράδειγμα μας, κάνουμε τη γραφική παράσταση της συνάρτησης f(x) = cos(x) - x^2 στο διάστημα [-π/3, π/3] με τη χρήση της συνάρτησης **Plot**:

```
Plot@Cos@xD - x^2, 8x, -π ê 3, π ê 3<, PlotStyle → RGBColor@1, 0, 0DD
```




```
¦ Graphics ¦
```

στην οποία χρησιμοποιήσαμε την επιλογή **PlotStyleØ**RGBColor[1,0,0] για να εμφανίσουμε τη γραμμή της καμπύλης με κόκκινο χρώμα. Από τη γραφική παράσταση πο προέκυψε, βρίσκουμε ότι τα σημεία τομής της με το άξονα των χ είναι τα σημεία με τετμημένη x @ -0,8 και x @ 0,8. Στη συνέχεια με αφετηρία της συνάρτησης **FindRoot** αυτές τις δύο τιμές βρίσκουμε τις προσεγγιστικές λύσεις της εξίσωσης:

```
FindRoot@Cos@xD - x^2 « 0, 8x, 0.8<D

8x → 0.824132<
```

```
FindRoot@Cos@xD - x^2 « 0, 8x, -0.8<D

8x → -0.824132<
```

5.2 Επίλυση συστημάτων

Οι βασικές συναρτήσεις που χρησιμοποιούμε για την επίλυση συστημάτων είναι οι **Solve** και **Reduce**, τις οποίες συντάσουμε κατά τα γνωστά, με μόνη διαφορά, ότι τις εξισώσεις του συστήματος είτε τις γράφουμε με μορφή λίστας είτε τις συνδέουμε με το λογικό τελεστή && (και) ενώ τους αγνώστους τους δηλώνουμε επίσης με μορφή λίστας {x, y, ...}.

Οι λύσεις (x_0, y_0, ...), (x_1, y_1, ...), ... παρουσιάζονται:

α) Σε λίστα υπό τη μορφή κανόνων αντικατάστασης:

$$\{\{x \to x_0, y \to y_0, ...\}, \{x \to x_1, y \to y_1, ...\}, ...\}$$

όταν χρησιμοποιούμε τη **Solve**. Για να τις εμφανίσουμε σε μορφή λίστας, χωρίς τους κανόνες αντικατάστασης,
μπορούμε κατά τα γνωστά να χρησιμοποιήσουμε τον τελεστή αντικταστάσης "/.".

β) Υπό τη μορφή:

$$x == x_0 \&\& y == y_0 \&\& ... || x == x_1 \&\& y == y_1 \&\& ... || ...$$

όταν χρησιμοποιούμε τη **Reduce**. Για να πάρουμε τις λύσεις σε μορφή λίστας εφαρμόζουμε πρώτα την εντολή {**ToRules**[%]} για να τις εμφανίσουμε ως κανόνες αντικατάστασης και στη συνέχεια εφαρμόζουμε
τον τελεστή αντικτάστασης {x, y, ...} /. %. Ισοδύναμα μπορούμε να εφαμόσουμε το σύνθετο τελεστή {x, y, ...} /. {**ToRules**[%]}.

Παράδειγμα 18: Έστω το σύστημα των δύο γραμμικών εξισώσεων

$$a x + b y = c$$
$$d x + e y = f$$

με δύο αγνώστους x και y. Επιλύνοντάς το με τη συνάρτηση **Solve** βρίσκουμε τις ακριβείς λύσεις του συστήματος

```
Solve@8a x + b y « c, d x + e y == f<, 8x, y<D

99x → -  -c e + b f                 , y → -  -c d + a f                 ==
          -b d + a e                          b d - a e
```

ενώ επιλύνοντάς το με τη συνάρτηση **Reduce** παίρνουμε μια πλήρη διερεύνηση του συστήματος

```
Reduce@a x + b y « c && d x + e y == f, 8x, y<D
```

$$x == \frac{-c\,e + b\,f}{b\,d - a\,e} \;\&\&\; y == \frac{c\,d - a\,f}{b\,d - a\,e} \;\&\&\; b\,d - a\,e \neq 0 \;\text{»»}$$
$$b == \frac{a\,e}{d} \;\&\&\; c == \frac{a\,f}{d} \;\&\&\; x == \frac{f - e\,y}{d} \;\&\&\; d \neq 0 \;\text{»»}$$
$$a == 0 \;\&\&\; c == \frac{b\,f}{e} \;\&\&\; d == 0 \;\&\&\; y == \frac{f}{e} \;\&\&\; e \neq 0 \;\text{»»}$$
$$d == 0 \;\&\&\; e == 0 \;\&\&\; f == 0 \;\&\&\; x == \frac{c - b\,y}{a} \;\&\&\; a \neq 0 \;\text{»»}$$
$$a == 0 \;\&\&\; b == 0 \;\&\&\; c == 0 \;\&\&\; d == 0 \;\&\&\; e == 0 \;\&\&\; f == 0 \;\text{»»}$$
$$a == 0 \;\&\&\; d == 0 \;\&\&\; e == 0 \;\&\&\; f == 0 \;\&\&\; y == \frac{c}{b} \;\&\&\; b \neq 0$$

5.2.1 Συστήματα με μία παράμετρο

Παράδειγμα 19: Έστω το σύστημα

$$k\,x^2 + y = 1$$
$$x + y = 1$$

όπου τα x και y είναι οι άγνωστοι και k παράμετρος του συστήματος.

Αρχικά δίνουμε ένα όνομα (sys1) στο σύστημα για να μπορέσουμε στη συνέχεια να αναφερόμαστε σε αυτό.

```
sys1 = 8k x^2 + y == 1, x + y « 1<
```

```
8k x^2 + y == 1, x + y == 1<
```

Στη συνέχεια επιλύνουμε το σύστημα (με τη χρήση της συνάρτησης **Solve**) και τη λύση την αποθηκεύουμε στην μεταβλητή sol1.

```
sol1 = Solve@sys1, 8x, y<D
```

```
98y → 1, x → 0<, 9y → -1 + k/k, x → 1/k==
```

Μπορούμε να κάνουμε επαλήθευση χρησιμοποιώντας τον τελεστή "/."

```
sys1 ê. sol1
```

```
98True, True<, 91/k + (-1 + k)/k == 1, 1/k + (-1 + k)/k == 1==
```

Παρατηρούμε ότι για το δεύτερο ζεύγος λύσεων η επαλήθευση δεν είναι σαφής. Στην περίπτωση αυτή μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **Simplify**:

```
sys1 ê. sol1 êê Simplify

88True, True<, 8True, True<<
```

Τέλος, μπορούμε να κάνουμε πλήρη διερεύνηση για τις διάφορες τιμές της παραμέτρου k με τη συνάρτηση **Reduce**:

```
Reduce@sys1, 8x, y<D

x == 0 && y == 1 »» x == 1/k && y == (-1 + k)/k && k ≠ 0
```

5.2.2 Συστήματα με δύο παραμέτρους

Παράδειγμα 20: Έστω το σύστημα

$$a\,x + b\,y = 1$$
$$x - b\,y = 2$$

όπου τα x και y είναι οι άγνωστοι και τα a, b οι παράμετροι του συστήματος.

Επιλύνουμε το σύστημα χρησιμοποιώντας τη συνάρτηση **Solve**:

```
Solve@8a x + b y « 1, x - b y « 2<, 8x, y<D

99x → 3/(1 + a), y → -(-1 + 2 a)/((1 + a) b)==
```

Κάνουμε επαλήθευση χρησιμοποιώντας τον τελεστή "/."

```
8a x + b y « 1, x - b y « 2< ê. %

99 3 a/(1 + a) - (-1 + 2 a)/(1 + a) == 1, 3/(1 + a) + (-1 + 2 a)/(1 + a) == 2==
```

Επειδή η επαλήθευση δεν είναι σαφής χρησιμοποιήσουμε τη συνάρτηση **Simplify**:

```
8a x + b y « 1, x - b y « 2< ê. %% êê Simplify

88True, True<<
```

Τέλος, κάνουμε πλήρη διερεύνηση για τις διάφορες τιμές των παραμέτρων a, b με τη συνάρτηση **Reduce**:

```
Reduce@8a x + b y « 1, x - b y « 2<, 8x, y<D

a == 1/2 && b == 0 && x == 2 »»
 x == 3/(1 + a) && y == (1 - 2 a)/((1 + a) b) && 1 + a ≠ 0 && b ≠ 0
```

5.2.3 Συστήματα βαθμού μεγάλυτερου του 2

Για τα συστήματα βαθμού n ¥ 2 ισχύουν τα ίδια ότι και για τα συστήματα πρώτου βαθμού.

Παράδειγμα 21: Έστω το σύστημα:

$$x^2 - y = 3$$
$$x - y = 2$$

Μπορούμε να βρούμε τις ακριβείς λύσεις του χρησιμοποιώντας τη συνάρτηση **Solve**:

```
Solve@8x^2 - y « 3, x - y « 2<, 8x, y<D

99y → -3/2 - Sqrt[5]/2, x → 1/2 I1 - Sqrt[5]M=, 9y → 1/2 I-3 + Sqrt[5]M, x → 1/2 I1 + Sqrt[5]M==
```

και να τις μετατρέψουμε σε αριθμητικές (προσεγγιστικές) λύσεις με τη συνάρτηση **N**:

```
N@%D

88y → -2.61803, x → -0.618034<, 8y → -0.381966, x → 1.61803<<
```

Επίσης τις αριθμητικές (προσεγγιστικές) λύσεις του συστήματος, μπορούμε και να τις βρούμε απευθείας χρησιμοποιώντας τη συνάρτηση **NSolve**:

```
NSolve@8x^2 - y « 3, x - y « 2<, 8x, y<D

88y → -0.381966, x → 1.61803<, 8y → -2.61803, x → -0.618034<<
```

5.2.4 Συστήματα με περισσότερους αγνώστους

Παράδειγμα 22: Έστω το παρακάτω σύστημα τριών εξισώσεων, με αγνώστους x, y, z:

$$x + 2y + z = 16$$
$$x - 2y + z = 0$$
$$2x + y - z = 5$$

Επιλύνουμε το σύστημα χρησιμοποιώντας τη συνάρτηση **Solve**:

```
Solve@8x + 2 y + z « 16, x − 2 y + z « 0, 2 x + y − z « 5<, 8x, y, z<D

88x → 3, y → 4, z → 5<<
```

Κάνουμε επαλήθευσηχρησιμοποιώντας τον τελεστή "/."

```
8x + 2 y + z « 16, x − 2 y + z « 0, 2 x + y − z « 5< ê. %

88True, True, True<<
```

Παράδειγμα 23: Έστω το σύστημα

$$\alpha x + y + z = 1$$
$$x + \alpha y + z = \alpha$$
$$x + y + \alpha z = a^2$$

όπου τα x, y, z είναι οι άγνωστοι και α παράμετρος του συστήματος.

Αρχικά δίνουμε ένα όνομα (sys2) στο σύστημα για να μπορέσουμε στη συνέχεια να αναφερόμαστε σε αυτό.

```
sys2 = 8a x + y + z « 1, x + a y + z « a, x + y + a z « a2<

8a x + y + z == 1, x + a y + z == a, x + y + a z == a2<
```

Επιλύνουμε το σύστημα (με τη χρήση της συνάρτησης **Solve**) και τη λύση την αποθηκεύουμε στην μεταβλητή sol2.

```
sol2 = Solve@sys2, 8x, y, z<D

99x → −(1 + a)/(2 + a), y → 1/(2 + a), z → −(−1 − 2 a − a^2)/(2 + a) ==
```

Κάνουμε επαλήθευση χρησιμοποιώντας τον τελεστή "/."

```
sys2 ê. sol2

99 1/(2 + a) − a H1 + aL/(2 + a) − (−1 − 2 a − a^2)/(2 + a) == 1,
  a/(2 + a) − (1 + a)/(2 + a) − (−1 − 2 a − a^2)/(2 + a) == a,
  1/(2 + a) − (1 + a)/(2 + a) − a H−1 − 2 a − a^2L/(2 + a) == a^2==
```

Επειδή η επαλήθευση δεν είναι σαφής χρησιμοποιήσουμε τη συνάρτηση **Simplify**:

```
sys2 ê. sol2 êê Simplify

88True, True, True<<
```

Τέλος, κάνουμε πλήρη διερεύνηση για τις διάφορες τιμές της παραμέτρου α με τη συνάρτηση **Reduce**:

```
Reduce@sys2, 8x, y, z<D

a == 1 && x == 1 - y - z »»
 x == -1 - a / (2 + a) && y == 1 / (2 + a) && z == (1 + 2 a + a^2) / (2 + a) && -1 + a ≠ 0 && 2 + a ≠ 0
```

5.2.5 Εύρεση προσεγγιστικών λύσεων

Φυσικά υπάρχουν συστήματα, τα οποία δεν μπορούν να επιλυθούν με ακρίβεια.

Παράδειγμα 24: Έστω το σύστημα

$$x^2 + \frac{1}{y^2} = 1$$
$$\frac{1}{x^4} - \frac{1}{y^6} = 2$$

Αρχικά δίνουμε ένα όνομα (sys3) στο σύστημα για να μπορέσουμε στη συνέχεια να αναφερόμαστε σε αυτό:

```
sys3 = 9x^2 + 1/y^2 « 1, 1/x^4 - 1/y^6 « 2=

9x^2 + 1/y^2 == 1, 1/x^4 - 1/y^6 == 2=
```

Επιλύνουμε το σύστημα (με τη χρήση της συνάρτησης **Solve**) και τη λύση την αποθηκεύουμε στην μεταβλητή sol3:

```
sol3 = Solve@sys3, 8x, y<D

99x → -, I-1 + Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D +
       2 Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D^2 -
       4 Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D^3 +
       Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D^4M,
  y → - "Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D=,
 9x → -, I-1 + Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D +
       2 Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D^2 -
       4 Root@1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 1D^3 +
```

```
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D^4M,
 y → "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D=,
9x → ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D +
      2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D^2 −
      4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D^3 +
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D^4M,
 y → − "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D=,
9x → ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D +
      2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D^2 −
      4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D^3 +
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D^4M,
 y → "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 1D=,
9x → − ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D +
      2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^2 −
      4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^3 +
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^4M,
 y → − "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D=,
9x → − ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D +
      2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^2 −
      4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^3 +
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^4M,
 y → "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D=,
9x → ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D +
      2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^2 −
      4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^3 +
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D^4M,
 y → − "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 2D=,
9x → − ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 3D +
      2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 3D^2 −
      4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 3D^3 +
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 3D^4M,
 y → − "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 3D=,
```

```
9x → -(1/3) (-1 + Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D +
       2 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^2 -
       4 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^3 +
       Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^4 M,
 y → (1/3) Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D =,
9x → (1/3) (-1 + Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D +
      2 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^2 -
      4 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^3 +
      Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^4 M,
 y → -(1/3) Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D =,
9x → (1/3) (-1 + Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D +
      2 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^2 -
      4 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^3 +
      Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D^4 M,
 y → (1/3) Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 3D =,
9x → -(1/4) (-1 + Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D +
       2 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^2 -
       4 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^3 +
       Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^4 M,
 y → (1/4) Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D =,
9x → (1/4) (-1 + Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D +
      2 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^2 -
      4 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^3 +
      Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^4 M,
 y → -(1/4) Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D =,
9x → (1/4) (-1 + Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D +
      2 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^2 -
      4 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^3 +
      Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D^4 M,
 y → (1/4) Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 4D =,
9x → -(1/5) (-1 + Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 5D +
       2 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 5D^2 -
       4 Root[1 - 2 #1 + #1^2 + 2 #1^3 - 4 #1^4 + #1^5 &, 5D^3 +
```

```
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^4M,
 y → − "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D=,
9x → − ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D +
      2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^2 −
      4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^3 +
      Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^4M,
 y → "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D=,
9x → ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D +
     2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^2 −
     4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^3 +
     Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^4M,
 y → − "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D=,
9x → ' I−1 + Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D +
     2 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^2 −
     4 Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^3 +
     Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D^4M,
 y → "Root@1 − 2 #1 + #1^2 + 2 #1^3 − 4 #1^4 + #1^5 &, 5D==
```

Παρατηρούμε ότι η συνάρτηση **Solve** μας επιστρέφει ρίζες χρησιμοποιώντας εκφράσεις της μορφής Root. Στην περίπτωση αυτή μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **NSolve**, η οποία μας επιστρέφει αριθμητικές (προσεγγιστικές) λύσεις:

```
NSolve@sys3, 8x, y, z<D
```

```
Solve::svars : Equations may not give solutions for all "solve" variables.
```

```
88x → −1.48849, y → 0. − 0.906993 ą<,
 8x → −1.48849, y → 0. + 0.906993 ą<,
 8x → −0.85559 − 0.684119 ą, y → −0.744341 − 0.411259 ą<,
 8x → −0.85559 − 0.684119 ą, y → 0.744341 + 0.411259 ą<,
 8x → −0.85559 + 0.684119 ą, y → −0.744341 + 0.411259 ą<,
 8x → −0.85559 + 0.684119 ą, y → 0.744341 − 0.411259 ą<,
 8x → −0.838151, y → −1.83339<, 8x → −0.838151, y → 1.83339<,
 8x → 0. − 0.667931 ą, y → −0.831565<,
 8x → 0. − 0.667931 ą, y → 0.831565<,
 8x → 0. + 0.667931 ą, y → −0.831565<,
 8x → 0. + 0.667931 ą, y → 0.831565<,
 8x → 0.838151, y → −1.83339<, 8x → 0.838151, y → 1.83339<,
 8x → 0.85559 − 0.684119 ą, y → −0.744341 + 0.411259 ą<,
 8x → 0.85559 − 0.684119 ą, y → 0.744341 − 0.411259 ą<,
 8x → 0.85559 + 0.684119 ą, y → −0.744341 − 0.411259 ą<,
 8x → 0.85559 + 0.684119 ą, y → 0.744341 + 0.411259 ą<,
 8x → 1.48849, y → 0. − 0.906993 ą<,
 8x → 1.48849, y → 0. + 0.906993 ą<<
```

Κάνουμε επαλήθευση χρησιμοποιώντας τον τελεστή "/." :

```
sys3 ê. %

88True, True<, 8True, True<, 8True, True<, 8True, True<,
 8True, True<, 8True, True<, 8False, False<, 8False, False<,
 8True, True<, 8True, True<, 8True, True<, 8True, True<,
 8False, False<, 8False, False<, 8True, True<, 8True, True<,
 8True, True<, 8True, True<, 8True, True<, 8True, True<<
```

```
9x^2 + 1/y^2, 1/x^4 - 1/y^6 = ê. %%

{{81. + 0. ą, 2. + 0. ą}, {81. + 0. ą, 2. + 0. ą},
 {81. - 4.44089 × 10^-16 ą, 2. + 3.88578 × 10^-16 ą},
 {81. - 4.44089 × 10^-16 ą, 2. + 3.88578 × 10^-16 ą},
 {81. + 4.44089 × 10^-16 ą, 2. - 3.88578 × 10^-16 ą},
 {81. + 4.44089 × 10^-16 ą, 2. - 3.88578 × 10^-16 ą}, {81., 2.},
 {81., 2.}, {81. + 0. ą, 2. + 0. ą}, {81. + 0. ą, 2. + 0. ą},
 {81. + 0. ą, 2. + 0. ą}, {81. + 0. ą, 2. + 0. ą}, {81., 2.},
 {81., 2.}, {81. + 4.44089 × 10^-16 ą, 2. - 3.88578 × 10^-16 ą},
 {81. + 4.44089 × 10^-16 ą, 2. - 3.88578 × 10^-16 ą},
 {81. - 4.44089 × 10^-16 ą, 2. + 3.88578 × 10^-16 ą},
 {81. - 4.44089 × 10^-16 ą, 2. + 3.88578 × 10^-16 ą},
 {81. + 0. ą, 2. + 0. ą}, {81. + 0. ą, 2. + 0. ą}}
```

Παρατηρούμε, ότι οι λύσεις που επιστρέφει η **NSolve** δεν επαληθεύουν ακριβώς το σύστημα, αλλά τείνουν να το επαληθεύσουν.

Τέλος υπάρχουν συστήματα, στα οποία ούτε η συνάρτηση **NSolve** δεν μπορεί να δώσει προσεγγιστικές λύσεις.

Παράδειγμα 25: Έστω το σύστημα

$$6\,\text{Cos}\,[x] = y$$
$$x = ą^y$$

Παρατηρούμε ότι η εξίσωση δεν μπορεί να λυθεί αριθμητικά με τη χρήση της συνάρτησης **NSolve**.

```
NSolve@86 Cos@xD « y, x « E^y<, 8x, y<D

– Solve::tdep : The equations appear to involve the
    variables to be solved for in an essentially non-algebraic way.

NSolve@86 Cos@xD == y, x == ą^y<, 8x, y<D
```

Σε τέτοιες περιπτώσεις, και όχι μόνο σε αυτές, χρησιμοποιούμε τη συνάρτηση **FindRoot**[eqns, {x, x_0}, {y, y_0}], η οποία αναζητεί μια αριθμητική λύση του συστήματος στην περιοχή του σημείου (x, y) = (x_0, y_0). Συγκεκριμένα, αναζητούμε μία αριθμητική λύση στη περιοχή του (x, y) = (‰, 1):

```
FindRoot@86 Cos@xD « y, x « E^y<, 8x, E<, 8y, 1<D

8x → 1.50285, y → 0.407363<
```

και μία αριθμητική λύση στην περιοχή του σημείου (x, y) = (‰^{2}, 2):

```
FindRoot@86 Cos@xD « y, x « Ey<, 8x, E2<, 8y, 2<D

8x → 7.51124, y → 2.0164<
```

5.2.6 Απαλοιφή αγνώστων από τις εξισώσεις συστήματος

Το *Mathematica* διαθέτει τη συνάρτηση

Eliminate[{eqns},{vars}]

η οποία μας επιτρέπει να απαλείψουμε τις αναγραφόμενες μέσα στη δεύτερη λίστα μεταβλητές μεταξύ των εξισώσεων του συστήματος.

Παράδειγμα 26: Έστω το σύστημα

a x + y = 1
x + (1-a) y = 2

Θεωρώντας το a ως παράμετρο και χρησιμοποιώντας τη συνάρτηση **Solve** βρίσκουμε τις ακριβείς λύσεις του συστήματος:

```
Solve@8a x + y « 1, x + H1 − aL y « 2<, 8x, y<D

99x → −(−1 − a)/(1 − a + a2), y → −(−1 + 2 a)/(1 − a + a2)==
```

Παρατηρούμε ότι μεταξύ των λύσεων του συστήματος υπάρχει ένας δεσμός, τον οποίο μπορούμε να βρούμε απαλείφοντας το a μεταξύ των δύο λύσεων. Αυτή ακριβώς, την εργασία κάνει η συνάρτηση **Eliminate**.

Εύρεση του δεσμού με τη χρήση της συνάρτησης **Eliminate**:

```
eqn2 = Eliminate@8a x + y « 1, x + H1 − aL y « 2<, aD

H1 − yL y == x2 + x H−2 + yL
```

Αν θέλουμε να επιλύσουμε τον δεσμό ως προς μία εκ των δύο μεταβλητών, μπορούμε ασφαλώς να χρησιμοποιήσουμε τη συνάρτηση **Solve**.

Επίλυση του δεσμού ως προς τη μεταβλητή x:

```
Solve@eqn2, xD

99x → 1/2 I2 − y − è(4 − 3 y2)M=, 9x → 1/2 I2 − y + è(4 − 3 y2)M==
```

Επίλυση του δεσμού ως προς τη μεταβλητή y:

```
Solve@eqn2, yD

99y → 1/2 I1 − x − è!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! 1 + 6 x − 3 x² M=, 9y → 1/2 I1 − x + è!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! 1 + 6 x − 3 x² M==
```

Παρατήρηση: Η συνάρτηση Solve μας παρέχει τη δυνατότητα να καταλήξουμε στο ίδιο αποτελέσματα. Συγκεκριμένα:

αν θέλουμε να λύσουμε το σύτημα ως προς τη μεταβλητή x απαλείφοντας την παράμετρο a, χρησιμοποιούμε την εντολή:

```
Solve@8a x + y « 1, x + H1 − aL y « 2<, x, aD

99x → 1/2 I2 − y − è!!!!!!!!!!!!!!!!!!!! 4 − 3 y² M=, 9x → 1/2 I2 − y + è!!!!!!!!!!!!!!!!!!!! 4 − 3 y² M==
```

ενώ αν θέλουμε να λύσουμε το σύτημα ως προς τη μεταβλητή y απαλείφοντας την παράμετρο a, χρησιμοποιούμε την εντολή:

```
Solve@8a x + y « 1, x + H1 − aL y « 2<, y, aD

99y → 1/2 I1 − x − è!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! 1 + 6 x − 3 x² M=, 9y → 1/2 I1 − x + è!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! 1 + 6 x − 3 x² M==
```

Κεφάλαιο 6ο: Όρια ακολουθιών και συναρτήσεων. Σειρές Taylor.

H entolή me thn opoía brískoume to ório miaV akolouqíaV eínai h Limit.

```
? Limit
```

```
Limit@expr, x->x0D finds the
   limiting value of expr when x approaches x0. More...
```

Me ??Limit **brískoume ta carakthristiká thV entolήV** Limit

```
?? Limit
```

```
Limit@expr, x->x0D finds the
   limiting value of expr when x approaches x0. More...
```

```
Attributes@LimitD = 8Listable, Protected<

Options@LimitD = 8Analytic → False, Direction → Automatic<
```

To protected **shmaínei óti den mporoúme na thn allάxoume enώ to** Listable **shmaínei óti mporeí na ejarmosteí sugcrónwV se pollέV akolouqíeV dhl. se mia olόklhrh lísta apo akolouqíeV. P.c**

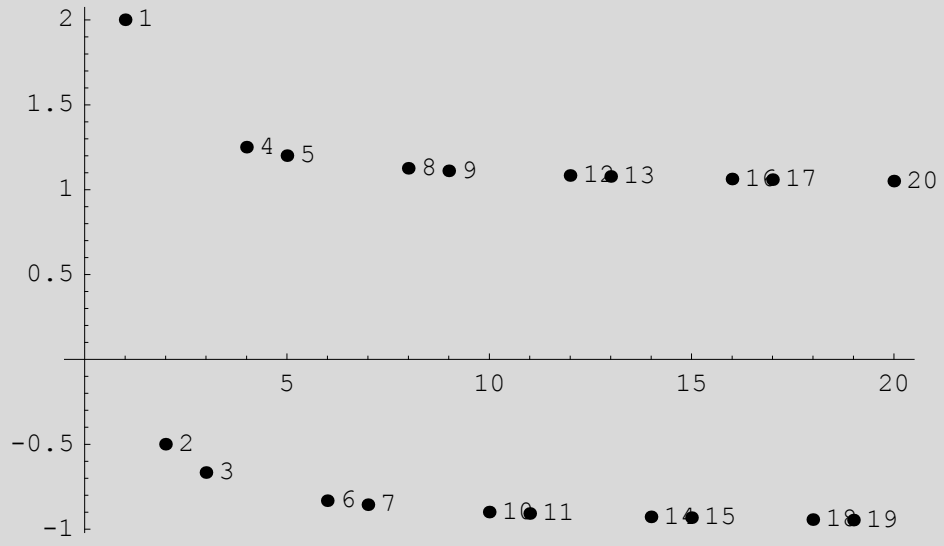
```
Limit@9 Log@nD/n, è!!n, 1 - n^3/n^2=, n → ∞D

80, 1, -∞<
```

Όpwn blέpoume h $\frac{\text{Log}[n]}{n}$ **eínai apoklínousa. Eάn h akolouqía écei dύo ή perissόtereV sugklínouseV upoakolouqíeV tóte apantάei me éna mήnuma thV morjήV** Interval[{a,b}] **pou shmaínei óti upάrcoun kάpoieV piqanέV lύseiV sto diάsthma** [**a**,**b**]. **p.c**

```
Limit@1/n + Sin@n π/2D + Cos@n π/2D, n → ∞D

Interval@8-2, 2<D
```

H parapάnw akolouqía écei duo oriakoύV ariqmoύV to 1 **kai to** -1. **ÁV kάnoume kai éna diάgramma na to doúme**:

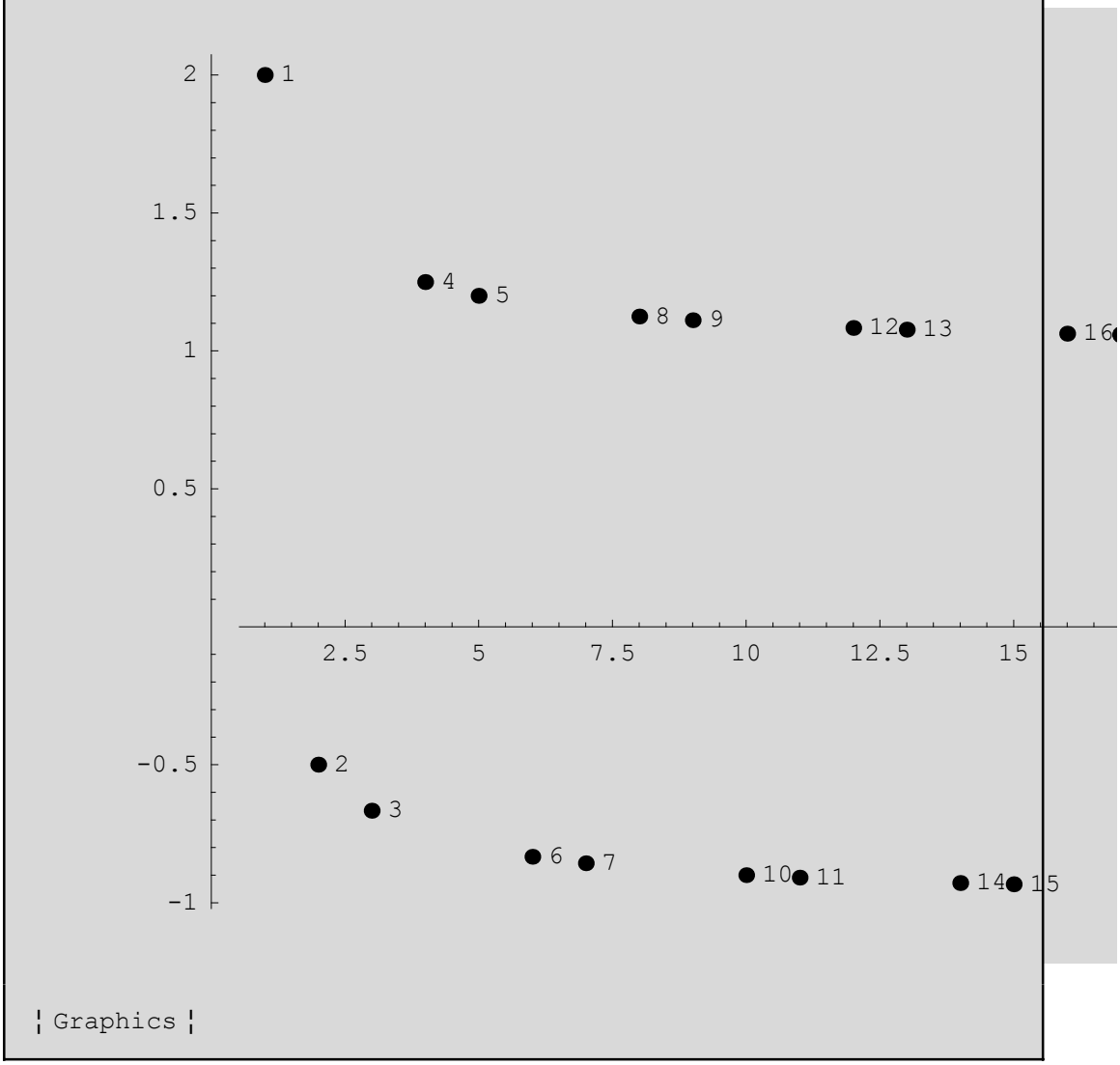
```
Remove@LabeledListPlotD

oiPrwtoi20oroi = TableA 1/n + SinA n π/2 E + CosA n π/2 E, 8n, 20<E;
<< Graphics`Graphics`
LabeledListPlot@oiPrwtoi20oroiD
```




```
¦ Graphics ¦
```

Epeidή den apeikonízontai ó la ta shmeía swstά, all άzoume ta carakthristikά thV LabeledListPlot:

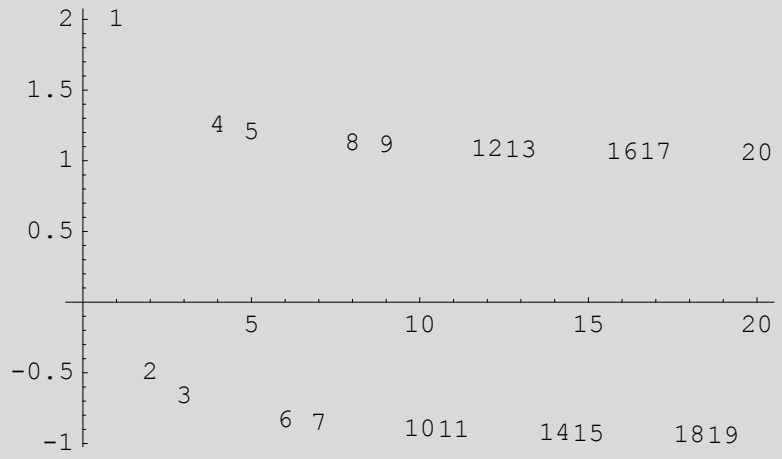
```
Remove@LabeledListPlotD

oiPrwtoi20oroi = TableA 1/n + SinA n π/2 E + CosA n π/2 E, 8n, 20<E;
<< Graphics`Graphics`
LabeledListPlot@oiPrwtoi20oroi,
 PlotRange → All, AxesOrigin → 80, 0<, AspectRatio → 0.8D
```




```
¦ Graphics ¦
```

Upάrcei kai άllοV énaV trópoV na doúme ta parapάnw shmeía me thn crήsh thV TextListPlot

```
TextListPlot[Table[1/n + Sin[n π/2] + Cos[n π/2], {n, 20}]]
```




```
-Graphics-
```

Upárcei kai mia ceiróterh períptwsh pou enώ upárcei to ório to Mathematica **adunateí na apantήsei ή na dώsei láqoV apánthsh p.c**

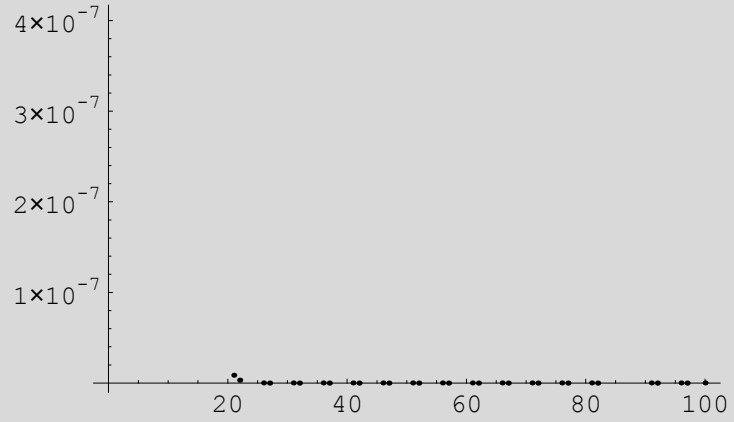
```
Limit[9n!/n^n, (n!)^(1/n)=, n → ∞]
```

```
Series::esss : Essential singularity encountered in Gamma[1/n + 1 + O[n]^3].
Series::esss : Essential singularity encountered in Gamma[1/n + 1 + O[n]^3].
Series::esss : Essential singularity encountered in Gamma[1/n + 1 + O[n]^3].
General::stop :
 Further output of Series::esss will be suppressed during this calculation.
```

```
9Limit[n^-n n!, n → ∞], Limit[(n!)^(1/n), n → ∞]=
```

Eínai gnwstó óti h prώth eínai mhdenikή kai h deúterh apoklínousa sto +¶.

```
ListPlotATable@8n, n^-n n!<, 8n, 100<D,
 PlotLabel -> "οι όροι της ακολουθίας n!/n^n"E
```

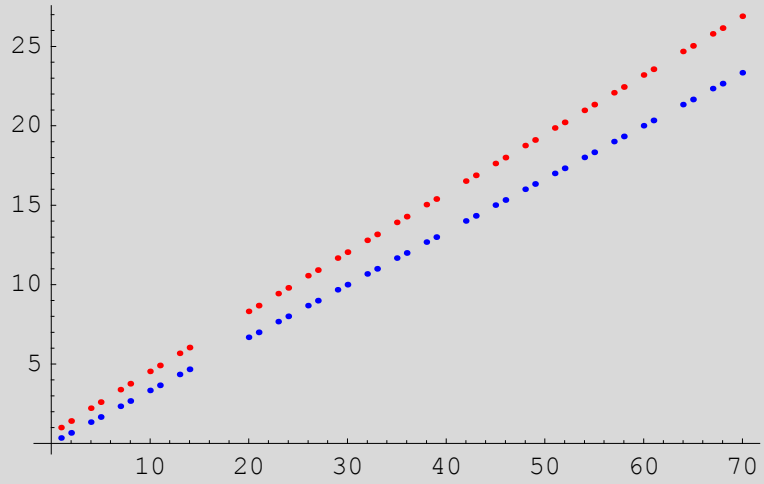



```
¦ Graphics ¦
```

AV scediάsoume tώra thn $n!/n^n$. **H** $n!/n^n$ **sumperijέretai grammikά me klίsh "perίpou" eίnai ίsh me** 1/3. **Gia na to apodeίxoume autό(grajikή apόdeixh!) scediάzoume kai thn** $1/3$ **me diajoretikά crώmata gia na doύme thn diajorά** :

```
timesToyn = 70
Remove@DisplayTogetherD
<< Graphics`Graphics`
p3 =
 DisplayTogetherAListPlotATableA9n, n!=, 8n, 1, timesToyn<E,
   PlotStyle → RGBColor@1, 0, 0DE, ListPlotA
   TableA9n, n/3=, 8n, 1, times<E, PlotStyle → RGBColor@0, 0, 1DEE
```

```
70
```




```
¦ Graphics ¦
```

Genikά h crήsh twn grajikών parastάsewn eínai éna crήsimo ergaleío. Όpou eínai ejiktó qa dínoume kai mia kattάllhlh grajikή parάstash.

6.1 Aqroísmata, Ginómena,SeirέV

Tó άqroisma dínetai apo thn sunάrthsh Sum. **Ta ória tou aqroísmatoV dínontai me kάpoieV timέV sta** min **kai** max. **To** d , **an upárcei paristánei to bήma pou auxánontai oi timέV thV metablhtήV** i. **SunoyízontaV écoume tiV parakάtw morjέV**

Sum[f,{i,max}] ή Sum[f,{i,min,max}] ή Sum[f,{i,min,max,d}]

Paradeígmata:

```
seira = SumA xi/i, 8i, 1, 10, 2<E
```

$$x+\frac{x^3}{3}+\frac{x^5}{5}+\frac{x^7}{7}+\frac{x^9}{9}$$

```
seira^2
```

$$\left(x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \frac{x^9}{9}\right)^2$$

```
Expand@seira^2D
```

$$x^2 + \frac{2\,x^4}{3} + \frac{23\,x^6}{45} + \frac{44\,x^8}{105} + \frac{563\,x^{10}}{1575} + \frac{124\,x^{12}}{945} + \frac{143\,x^{14}}{2205} + \frac{2\,x^{16}}{63} + \frac{x^{18}}{81}$$

Όpwv blépoume mia seirá mporoúme thn ceiristoúme ópwV qéloume. P.c na broúme to tetrágwnó thV kai epíshV mporoúme kai na thn paragwgήsoume ή na oloklhrώsoume!

```
D@seira, 8x, 2<DH  deyterh paragwgos ws pros x  L
integ = Integrate@seira, xDH  aoristo oloklhrwma ws pros x L
```

$$2\,x + 4\,x^3 + 6\,x^5 + 8\,x^7$$

$$\frac{x^2}{2} + \frac{x^4}{12} + \frac{x^6}{30} + \frac{x^8}{56} + \frac{x^{10}}{90}$$

H sunárthsh Coefficient **eínai polú crήsimh dióti ótan to anáptugma thV seiráV eínai arketá megálo mporeí na maV breí ton suntelestή miaV dúnamhV x^n miaV metablhtήV** x . **P.c**

```
Coefficient@integ, x, 10D
```

$$\frac{1}{90}$$

Álleς paralagéV tou Coefficient **mporeíte na breíte patώntaV** F1 **kata ta gnwstá. Parakátw paraqétontai ta merikά aqroísmata dunámewn tou** i **se morjή pínaka kai h gnwstή maV gewmetrikή pródo me lógo w**.

```
                n
TableA9m,  ,  i^m=, 8m, 1, 3<E êê TableForm
               i=1
```

1	$\frac{1}{2}\,n\,(1 + n)$
2	$\frac{1}{6}\,n\,(1 + n)\,(1 + 2\,n)$
3	$\frac{1}{4}\,n^2\,(1 + n)^2$

```
Sum@a ω^(i-1), 8i, 1, n<D
```

$$\frac{a\,(-1+\omega^{n})}{-1+\omega}$$

Gia na broúme an h parapánw gewmetrikή seirá sugklínei qétoume n **= ¶**

```
Sum@a ω^(i-1), 8i, 1, ∞<D
```

$$-\frac{a}{-1+\omega}$$

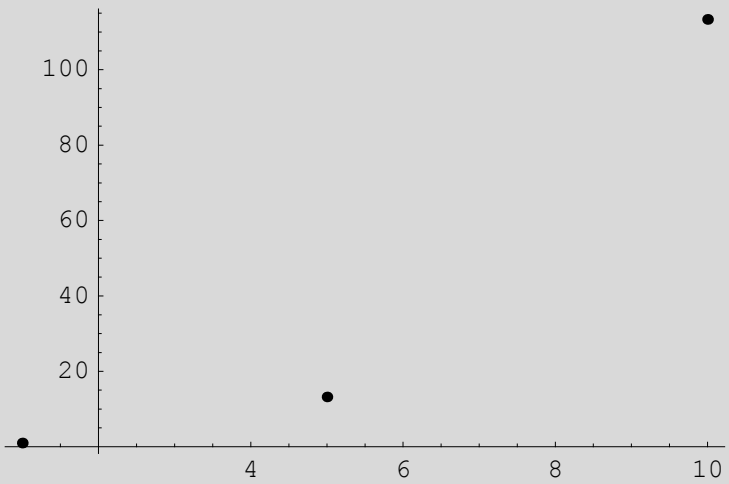
Prépei na prosέxoume óti siwphlά h Sum **upoqétei óti o paranomastήV** -1+**w écei thn idiόthta** |**w**|<1 **diόti alliώV den sugklínei**! **AV doúme ti gínetai se antíqeth períptwsh**

```
Sum@H1.5L^(i-1), 8i, 1, ∞<D
```

```
-2.
```

pou jusikά eínai lάqoV(diόti ejarmózetai lάqoV o túpoV $-\frac{1}{1+\omega}$ **gia w**=1.5). **AV doúme lígo kai thn grajikή parάstash twn merikών aqroismάtwn gia na to epibebaiώsoume**

```
ListPlot@Table@8n, Sum@H1.5L^(i-1), 8i, 1, n<D<, 8n, 10<D,
 PlotStyle → AbsolutePointSize@4DD
```




```
¦ Graphics ¦
```

To sύmbolo tou apeírou mporeί na eisacqeί grάjontaV \[Infinity] (**metaxύ tou** Infinity **kai**] **den prépei na upάrcei kenό gιatί den qa metarapeί autόmata se ¶**) ή **patώntaV to plήktro** Esc **kai metά**

grájontaV inf **kai xanapatώntaV to** Esc **crhsimopoiώntaV thn basikή palέta** 3 BasicInput. Όmoia **to** $/_{n=0}^{m}$ **mporeí na grajeí patώntaV ta plήktra** sum + n=0 % m â 1 /f[n] â (**ta** [**kai**] **den ta ktupáme! to** +]**gia parádeigma shmaínei óti patáme mazí to kai to** + **enώ to** â **eínai to mazí me to** SPACE **plήktro**) . **To** Limit **adunateí na breí ómwV thn ápeirh seirά ótan upárcei kάpoia staqerά**:

```
Limit@Sum@aω^(i-1), 8i, 1, n<D, n → ∞D
```

$$\text{Limit}\left(\frac{-1+a\omega^{n}}{-1+a\omega}, n \to \infty\right)$$

Gia na broúme to áqroisma thV seirάV me Limit **qa prépei angkastikά na báloume kάpoieV timέV sta** a **kai w p.c**

```
Limit@Sum@-1 + 2 H-1L^n, 8n, 1, m<D, m → ∞D
```

$$\text{Limit}\left(\frac{-1+2\,(-1)^{n}}{-1+2\,(-1)^{n}}, n \to \infty\right)$$

```
5
```

$$5$$

Me Sum[f,{i,n,m},{j,k,l}] **mporoúme na pároume aqroísmata thV** f [i,j] **ótan écoume pάnw apo mia metablhtή(edώ écoume thn** i **kai thn** j) **p.c**.

```
Sum@x^i y^j, 8i, 1, 4<, 8j, 1, 2<D
```

$$x\,y + x^2\,y + x^3\,y + x^4\,y + x\,y^2 + x^2\,y^2 + x^3\,y^2 + x^4\,y^2$$

H basikή entolή pou upologίzoume ginόmena eínai h

```
Product[f,{i,imin,imax}]
```

Mporoúme na broúme kai diplά ginόmena ή **ta όria gia** imax->¶ **όpwV gínetai kai me to** Sum. **Paradeígmata**:

```
Product@H i + 1L^2 ê i Hi + kL, 8i, 1, n<D
```

$$\text{Product}\left(\frac{(i+1)^2}{i\,(i+k)}, i, 1, n\right)$$

```
Product@H i + 1L^2 ê i Hi + kL, 8i, 1, ∞<D
```

$$\text{Product}\left(\frac{(i+1)^2}{i\,(i+k)}, i, 1, \infty\right)$$

```
H1 + nL Gamma@1 + kD Gamma@2 + nD ê Gamma@1 + k + nD
```

$$\frac{(1+n)\,\Gamma(1+k)\,\Gamma(2+n)}{\Gamma(1+k+n)}$$

```
Gamma@1 + kD
```

$$\Gamma(1+k)$$

Άskhsh: **H sunάrthsh j**(x) **tou** Euler **dίnei to plήqoV twn akeraίwn** n **metaxύ tou** 1 **kai tou** x **oi opoίoi den έcoun kanέna koinό diairέth me ton** x. **Sto** *Mathematica* **h sunάrthsh autή dίnetai me thn** EulerPhi[x] **p.c**

```
EulerPhi@60D

16
```

Sth Qewrίa Ariqmών dίnetai me ton tύpo j HxL = *x* ‰ I1 - 1/*p* Mή **me άlla lόgia eίnai to** x **epί to ginόmeno** (*p*ê*x*) **twn όrwn** 1 - 1/*p* **όpou to** p **eίnai έnaV prwtoV diairέthV tou** x **mikrόteroV jusikά apo to** x. **Na orίsetai mia sunάrthsh**
phiEuler[x_Integer] **pou qa eίnai akribώV to parapάnw ginόmeno Elέgxte ta apotelέsmata saV me thn boήqeia thV** EulerPhi.

6.2 Όria sunartήsewn miaV metablhtήV

Έ**stw** f(x) **mia sunάrthsh kai qέloume na upologίsoume to όrio thV kaqώV to** x **teίnei sto** x_0. **Autό grάjetai sto** *Mathematica* Limit@*f*@*x*D, *x* -> x_0]. **P.c**

```
Limit[(9 x^2 - 1)/(9 x^2 + 51 x - 18), x → 1/3]

2/19
```

Ό**tan to όrio den upάrcei ή adunateί na to breί, tόte bgάzei to mήnuma I**nterval[{a,b}] ή **kάti άllo p.c**

```
Limit[Sin[1/x], x → 0]

Interval@8-1, 1<D
```

To Interval[{-1,1}] **maV bebaiώnei όti sίgoura den upάrcei to όrio kai όti ίswV upάrcoun όria upoakolouqiών metaxύ tou** -1 **kai** 1. **Genikά όtan douleύomai me ta όria prέpei na eίmaste prosektikoί. Upάrcei dustucώV kai h perίptwsh pou to** *Mathematica* **dίnei lάqoV όrio. Gia parάdeigma aV pάroume thn** *f*@*x*D = (‰*x*-2‰)/(*x*-2) **pou jusikά paίrnei timέV** +1 **apo dexiά kai** -1 **apό aristerά. Opόte to όrio** Limit[f[x],x->2] **den qa έprepe na upάrcei. To** *Mathematica* **όmwV dίnei lάqoV apάnthsh**:

```
                Abs@x - 2D
f@xD := iiiiiiiiiiiiiiiiiiiiii
                  x - 2
Limit@f@xD, x → 2D
```

```
1
```

Gia touV parapánw lógouV qa prépei na paírnoume merikéV jorév kai ta pleurikά ória gia na diapistώnoume an parousiάzetai kάpoia asunéceia ή όci.To όrio apo ta aristerά to paírnoume me Direction->1 enώ apo ta dexiά me Direction->-1.

```
Limit@f@xD, x → 2, Direction → 1D
Limit@f@xD, x → 2, Direction → -1D
```

```
-1
```

```
1
```

6.3 Diplά ória sunartήsewn dύo metablhtών

To Mathematica **maV parécei thn dunatόthta eύreshV mόno twn diplών orίwn** Limit[Limit[f[x,y],y->y0],x->x0] **kai** Limit[Limit[f[x,y],x->x0],y->y0]. **Den maV parécetai h dunatόthta na broύme to** όrio Limit[f[x,y],(x,y)->(x0,y0)]. **Autά ta όria eínai crήsima gia na bgάloume kάpoia sumperάsmata gia thn sumperijorά miaV sunάrthshV kontά sto shmeίo** (x0,y0). **Gia parάdeigma an ta όria** Limit[-Limit[f[x,y],y->y0],x->x0] **kai** Limit[Limit[f[x,y],x->x0],y->y0] **όtan upάrcoun allά den eínai ίsa tόte den mporeί na upάrcei tό diplό όrio**. **Antίstroja an upάrcei to όrio tόte ta diplά eínai ίsa**. **AV pároume gia parάdeigma thn** f[x,y]= **kai aV broύme ta diplά όria sto** (0,0):

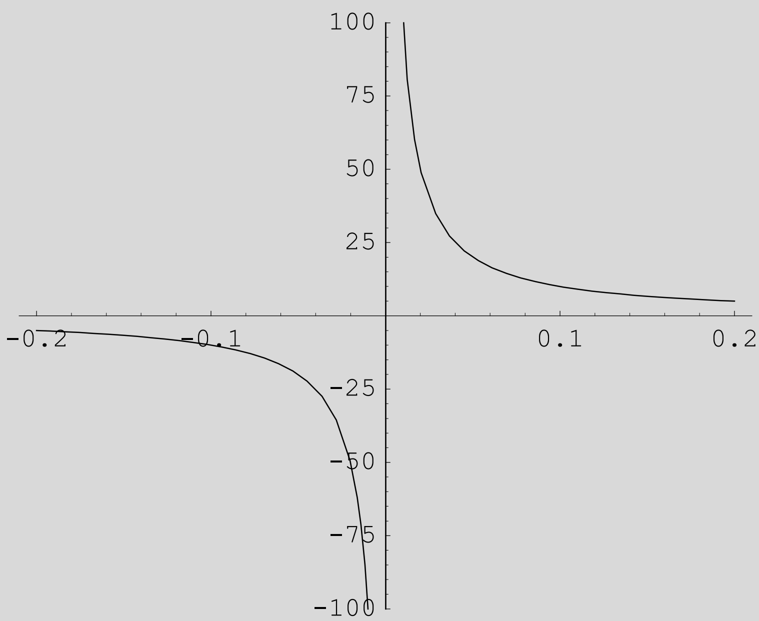
```
Remove@f, x, yD
                  x^2 + y
f@x_, y_D := iiiiiiiiiiiiiiiiiiiii
                x^2 + y^2
Limit@Limit@f@x, yD, x → 0D, y → 0D
Limit@Limit@f@x, yD, y → 0D, x → 0D
```

```
∞
```

```
1
```

Ta pleurikά ória kai oi grajikέV parastάseiV mporoύn na maV bohqήsoun na katanoήsoume kalύtera thn sumperijorά thV f **gύrw apo éna shmeίo** (x0,y0). **Edώ dίnontai kάpoia paradeígmata me thn** Plot. **An duskoleύeste kai zhtάte boήqeia**, **mporeίtai na metajέretai ton kέrsora páw se mia sunάrthsh** ή **se kάpoia epilogή thV**(**p.c páw sthn** Plot) **kai metά patήste** F1 **gia na párete boήqeia kai paradeígmata scetikά me thn sunάrthsh pou qélete boήqeia**.

```
Remove@fD
                               x^2 + y
f@x_, y_D := IfAx^2 + y^2 ≠ 0, iiiiiiiiiiiiiiiiiii , 0E
                              x^2 + y^2
H  αν ο παρανομαστής είναι 0 τότε θέτουμε τιμή =0 L
x = 0;
                          x^2 + y
ymin = −.2; ymax = .2; iiiiiiiiiiiiiiiiiii
                         x^2 + y^2
Plot@f@x, yD, 8y, ymin, ymax<, PlotRange → 8−100, 100<,
 PlotPoints → 50, AxesOrigin → 80, 0<,
 AspectRatio → .8, MaxBend → 20, Compiled → FalseD

1
iii
y
```



¦ Graphics ¦

Bálame skópima perissótera carakthristiká (options) sthn Plot apo ósa prágmati creiázontai ώste na mporeí kápoioV na máqei plhrojoríeV gia autá mésw tou plήktrou F1. Basiká creiázetai h plhrojoría PlotRangeØ{-100,100} giatí cwríV autή íswV den saV bgeí ena katanohtó grájhma. Me thn PlotRange kóboume katá boúlhsh ton áxona twn x ή twn y. (edώ o 0y scediázetai gia timéV -100 έwV 100 kai me AspectRatioØ.8 zhtáme to mήkoV tou káqetou áxona na eínai to 8/10 tou mήkouV tou orizóntiou). Apó to scήma parathroúme óti gia mikréV timéV tou x kontá sto 0 p.c x=0 ta pleuriká ória kaqώV y->0 eínai ¶ kai -¶ antístoica. AV to doúme thn apánthsh me to Limit:

```
Limit@Limit@f@x, yD, x → 0D, y → 0, Direction → 1D
Limit@Limit@f@x, yD, x → 0D, y → 0, Direction → −1D
```

```
LimitALimitAIfAx^2 + y^2 ≠ 0, ⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈ, 0E, y → 0E, x → 0, Direction → 1E
                              x^2 + y^2
```

```
LimitALimitAIfAx^2 + y^2 ≠ 0, ⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈ, 0E, x → 0E, y → 0, Direction → −1E
                              x^2 + y^2
```

Den ta katάjere! **Edώ den jtaίei to** Limit **allά o orismόV me to** If **pou dώsame gia thn** f. **To** Limit **den** έ**cei prόblhma me mhdenikoύV paranomastέV h** Plot ό**mwV mporeί na** έ**cei lόgw tou trόpou pou scediάzei thn grajikή parάstash**(**paίrnei kάpoia shmeία ston άxona O**x **brískei ta timέV thV** f **kai enwnei me euqόgramma tmήmata ta shmeία pou prokόptoun**!). **AV antikatastήsoume loipόn thn** f **me to klάsma** ⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈ **mέsa sto** Limit:
x^2+y^2

```
                                x^2 + y
LimitALimitA ⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈ, x → 0E, y → 0, Direction → 1E
                               x^2 + y^2
                                x^2 + y
LimitALimitA ⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈⅈ, x → 0E, y → 0, Direction → −1E
                               x^2 + y^2

−∞
```

```
∞
```

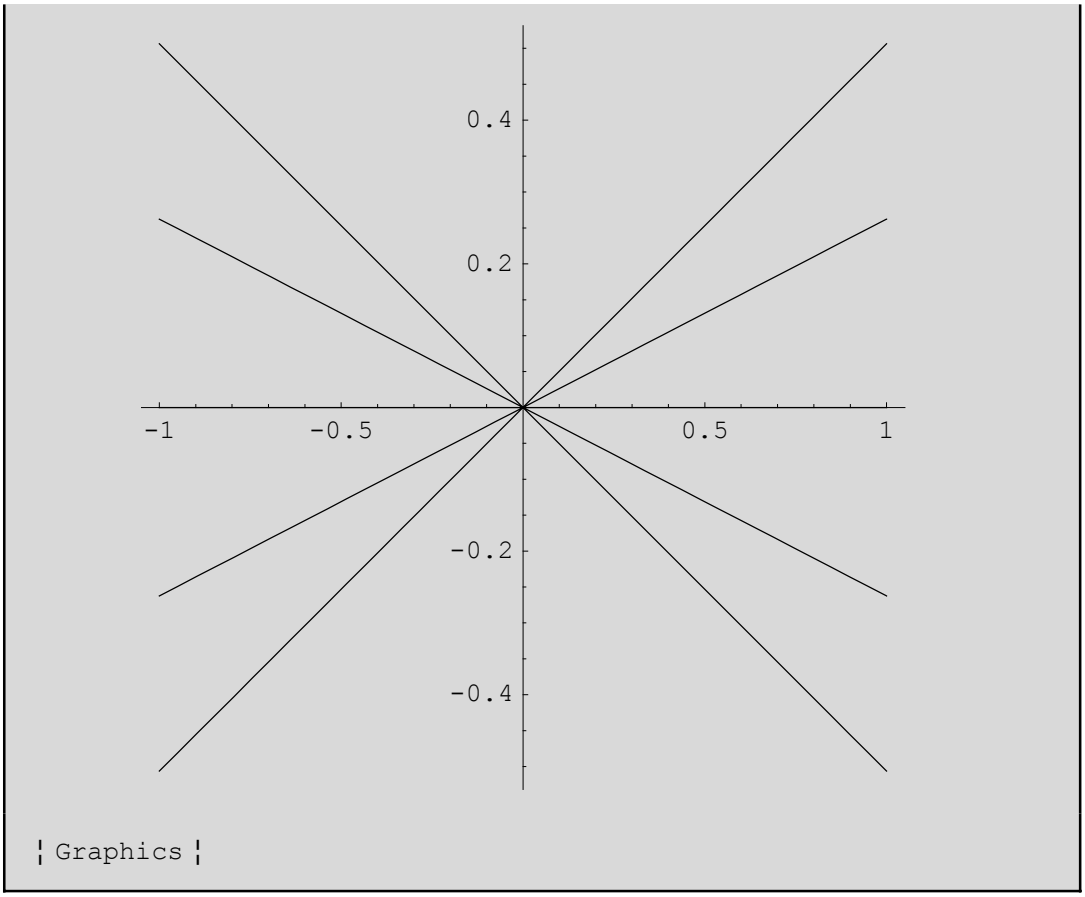
Sto *Mathematica* **upάrcei kai h perίergh periptwsh na upάrcei to** όrio **kai na** "**mhn upάrcoun**" **ta diplά** ό**ria**. Έ**na tέtoio kakό parάdeigma eίnai h sunάrthsh**
f[x,y]=x*Sin(1/y). **T**o é**na apo ta diplά** ό**ria eίnai to** Interval[{0,0}](**dhl**. **ousiastikά** ό**rio to** 0!!!):

```
Limit@Limit@x Sin@1 ê yD, x → 0D, y → 0D
Limit@Limit@x Sin@1 ê yD, y → 0D, x → 0D

0
```

```
Interval@80, 0<D
```

AV kάnoume kai edώ thn grajikή parάstash gia arketέV timέV tou y **kontά sto** 0 **gia na katalάboume thn apάnthsh** Interval[{0,0}].

```
Remove@x, y, z, gD;
g@x_, y_D := If@y ≠ 0, x Sin@1 ê yD, 0D
xmin = −1; xmax = 1;
pinakas = Table@g@x, yD, 8y, −.02, .02, 1 ê 100<D
Plot@Evaluate@pinakasD, 8x, xmin, xmax<,
 PlotRange → All, AspectRatio → 1, PlotPoints → 40D

80.262375 x, 0.506366 x, 0, −0.506366 x, −0.262375 x<
```



apo autá blépoume óti oi timéV tou eswterikoú oríou Limit[x Sin[1/y],y**Ø**0] **den teínoun se éna ório** f[x] **dhladή se kápoia sunárthsh tou** x (**blépe kai** pinakas)**me apotélesma na eínai "adúnato" na breqeí kai to exwterikó ório** Limit[Limit[x Sin[1/y],y**Ø**0],x**Ø**0]. **Étsi állote paírnoume ório** 0 **apo dexiá** H0+L **kai állote** 0 **apo aristerá**H0-L.**To** Evaluate **metá thn** Plot **anagkázei na upologistoún prώta óleV oi sunartήseiV tou** pinaka **prin ejarmosteí h** Plot. **Diajoretiká h** Plot **den qa mporoúse na kánei thn grajikή parástash díoti qa nómize óti o** pinaka **eínai pínakaV kai óci kápoieV sunartήseiV**!)

Άskhsh: Ajoú breíte apo to Help **tou** *Mathematica* **ti kánoun oi parakátw sunartήseiV na ermhneúsetai me tiV grajikéV parastáseiV pou parágontai**, **thn sumperijorá twn diplών oríwn thV** g[x_,y_]:=If[y**ƒ**0,x Sin[1/y],0]. (**prosocή an patήsete diplό klik se kápoia apo tiV grajikéV parstáseiV pou qa prokúyoun qa deíte** Movie!)

```
<< Graphics`Animation`
g@x_, y_D := If@y ≠ 0, x Sin@1 ê yD, 0D
MoviePlot@g@x, yD, 8y, -1, 1<,
 8x, -1, 1, 1 ê 20<, PlotRange → 8-1, 1<D
MoviePlot@g@x, yD, 8x, -1, 1<,
 8y, -1, 1, 1 ê 20<, PlotRange → 8-1, 1<D
```

6.4 Akrótata sunartήsewn

Gia na broúme éna topikó eláciston thV f[x] **górw apo to** x_0 **grájoume** FindMinimum@ f@xD, 8x, x_0}]. **Gia to topikó mégisto arkeí na zhtήsoume to eláciston thV** -f[x] **dióti** maxf= -min[-f]. **An bébaia écoume sunάthsh perisóterwn metablhtών tóte gínontai oi katάllhleV allagéV p.c gia na broúme to topikó eláciston górw apó to shmeío** Hx0, y0L **grájoume** Fin dMinimum@ f@x, yD, 8x, x0<, 8y, y0}]. **AV doúme tiV sunartήseiV autéVse scésh me thn** g **pou ήdh meletήsame**:

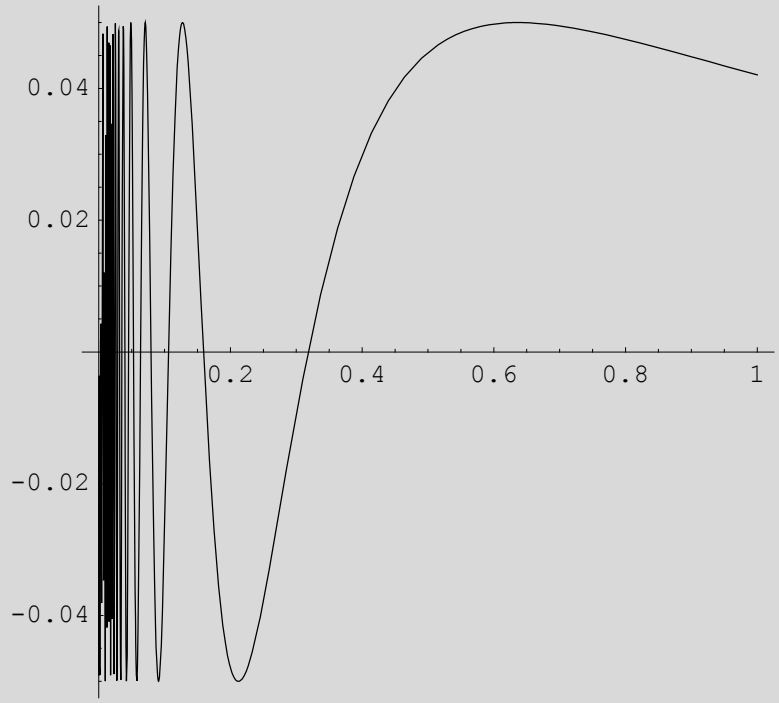
```
FindMinimum@g@.05, yD, 8y, 0.5<D
-FindMinimum@-g@.05, yD, 8y, 0.5<D
```

```
8-0.05, 8y → 0.212207<<
```

```
80.05, 8-Hy → 0.63662L<<
```

Dhladή gia x0=0.05 é**coume mégisto** 0.05 **gia** y=0.63662 **kai eláciston**-0.05 **gia** y=0.212207.A**V kánoume kai thn grajikή parάstash**:

```
ymin = 0; ymax = 1;
Plot@g@.05, yD, 8y, ymin, ymax<,
 PlotRange → All, AspectRatio → 1, PlotPoints → 40D
```




```
¦ Graphics ¦
```

An prospaqήsoume na broúme mégisto kai elάcisto thV g **gia timέV tou** y **se éna diάsthma** [ymin,ymax] **qa prépei na bάloume ta ória pou qa kiniétai to** y **wV exhV**

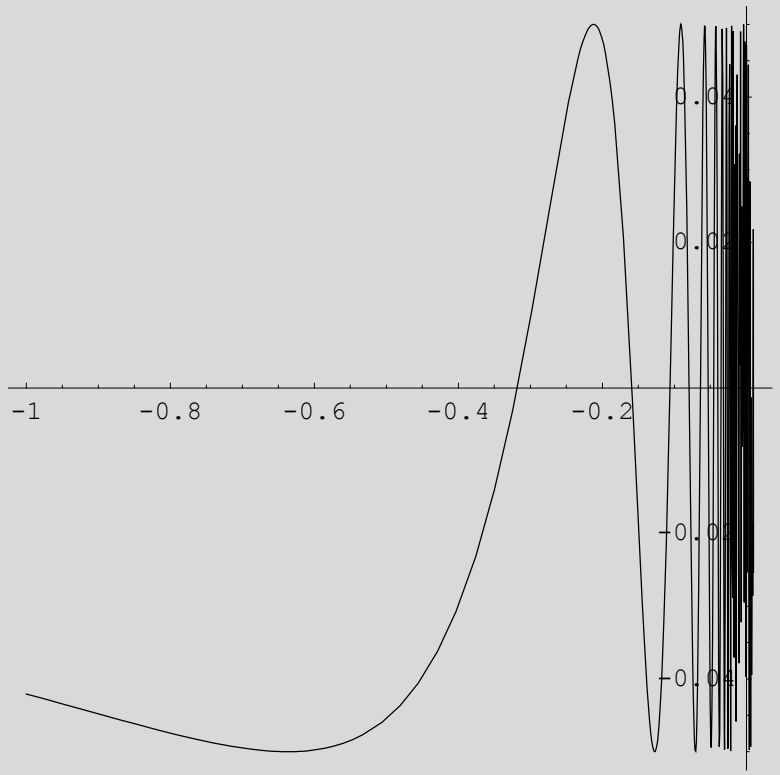
```
y0 = 0.005; ymin = −1; ymax = .01;
FindMinimum@g@0.05, yD, 8y, y0, ymin, ymax<D
−FindMinimum@−g@0.05, yD, 8y, y0, ymin, ymax<D
```

```
8−0.05, 8y → 0.00501275<<
```

```
80.05, 8−Hy → 0.00630317L<<
```

Apo autά blépoume óti h eύresh tou topikoú megístou kai elacístou exartάtai apo to shmeío pou tou dínomai. Έtsi **gia** y_0 =0.5 έ**dwse** ά**lla akr**ό**tata kai gia** y_0 =0.005 ά**lla**.

```
Plot@g@0.05, yD, 8y, ymin, ymax<,
 PlotRange → All, AspectRatio → 1, PlotPoints → 40D
```




```
¦ Graphics ¦
```

An tώra staqeropoiήsoume thn timή tou y **p.c** y=.05 **blépoume óti eínai adúnaton na pároume kápoia akrótata**:

```
x0 = 0.05; xmin = −1; xmax = 1;
FindMinimum@g@x, .05D, 8x, x0, xmin, xmax<D
```

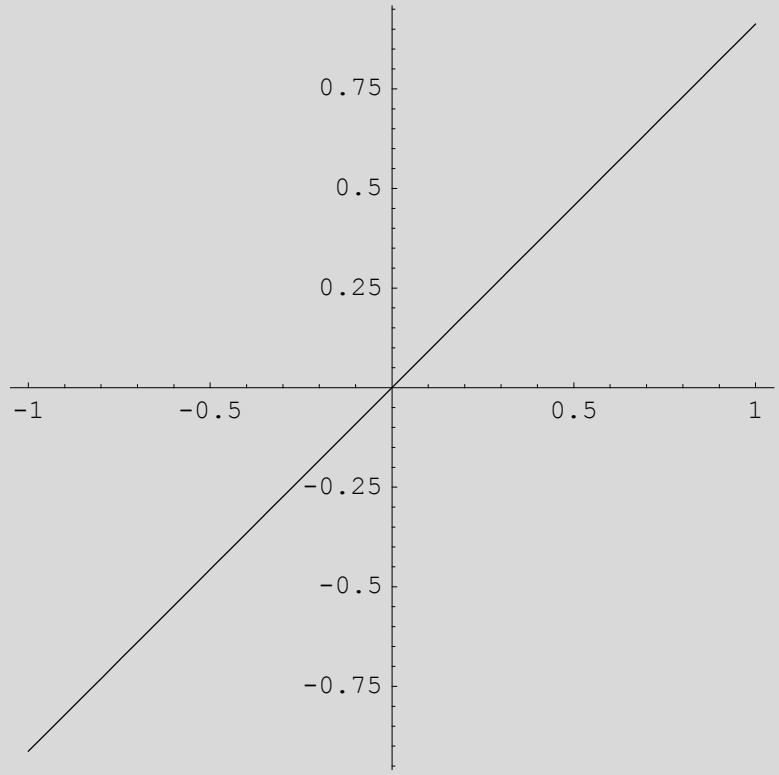
```
− FindMinimum::regex :
   Reached the point 8−1.64894< which is outside the region 88−1., 1.<<.
```

```
FindMinimum@g@x, 0.05D, 8x, x0, xmin, xmax<D
```

Dhladή den mpórese na breí éna topikó eláciste sta doqéntaória tou x . **AV kánoume páli thn grajikή parástash gia na doúme ti akribώV sumbaínei**:

```
Remove@xD
g@x, 0.05D
Plot@g@x, 0.05D, 8x, xmin, xmax<,
 PlotRange → All, AspectRatio → 1, PlotPoints → 40D
```

```
0.912945 x
```




```
¦ Graphics ¦
```

prágmati loipón den écei kápoio akrótato allá oi timéV g[x,0.05] **megalónoun ή mikraínoun sunéceia**! **An pároume kai ta akróta kai wV proV tiV dúo metablhtéV qa diapistώsoume ta ídia**:

```
x0 = 0.05; xmin = −1; xmax = 1; y0 = 0.005; ymin = −1; ymax = .01;
FindMinimum@g@x, yD, 8x, x0, xmin, xmax<, 8y, y0, ymin, ymax<D
```

```
– FindMinimum::fmgs :
   Could not symbolically find the gradient of g@x, yD. Try using
     the default method, giving two starting values for each variable.
```

```
FindMinimum@g@x, yD, 8x, x0, xmin, xmax<, 8y, y0, ymin, ymax<D
```

6.5 SeirέV dunάmewn, SeirέV Taylor kai Mac-Laurin

Έstw f(x) sunάrthsh, pou écei suneceíV paragώgouV wV proV x mécri kai n tάxewV sto shmeío a kai upάrcei h parάgwgoV n+1 tάxhV thV f sto a tóte upάrcei to anάptugma thV f se seirά gύrw apo to a se dunάmeiV $(x-a)^n$ p.c.gia n=4

```
Series@f@xD, 8x, a, 4<D
```

$$f(a)+f'(a)(x-a)+\frac{1}{2}f''(a)(x-a)^2+\frac{1}{6}f^{(3)}(a)(x-a)^3+\frac{1}{24}f^{(4)}(a)(x-a)^4+O(x-a)^5$$

To n+1=5 kai to $O(x-a)^5$ paristάnei to upóloipo(ή sjάlma) n+1 tάxhV thV f gia to shmeío a kai écei thn idiόthta to όrio kaqώV x->a na eínai 0 dhl. Limit[$O(x-a)^5$,x->a]=0. O parapάnw tύpoV eínai o tύpoV tou Taylor gia thn f. Eidikά όtan a=0 o tύpoV autόV gínetai

```
Series@f@xD, 8x, 0, 4<D
```

$$f(0)+f'(0)x+\frac{1}{2}f''(0)x^2+\frac{1}{6}f^{(3)}(0)x^3+\frac{1}{24}f^{(4)}(0)x^4+O(x)^5$$

pou den eínai άlloV apo ton tύpo tou Mac Laurin gia thn f. Allά aV doύme kai paradeígmata:

```
seira = Series@Log@xD, 8x, 1, 4<D
```

$$(x-1)-\frac{1}{2}(x-1)^2+\frac{1}{3}(x-1)^3-\frac{1}{4}(x-1)^4+O(x-1)^5$$

Thn parapάnw seirά mporoύme na thn uyώsoume sto tetrάgwno na thn paragwgίsoume wV proV x na thn oloklhrώsoume p.c

```
seira^2
D@seira, 8x, 2<D η δευτερη παράγωγος ως προς x  L
Integrate@seira, xD αόριστο ολοκληρωμα ως προς x  L
```

$$(x-1)^2-(x-1)^3+\frac{11}{12}(x-1)^4-\frac{5}{6}(x-1)^5+O(x-1)^6$$

$$-1+2(x-1)-3(x-1)^2+O(x-1)^3$$

$$\frac{1}{2}(x-1)^2-\frac{1}{6}(x-1)^3+\frac{1}{12}(x-1)^4-\frac{1}{20}(x-1)^5+O(x-1)^6$$

Χρήσιμες και στενά συνδεμένες με την Series είναι και οι Normal που αποκόπτει το υπόλοιπο O@x – aD^n απο το ανάπτυγμα Taylor και η SeriesCoefficient που επιστρέφει τον συντελεστή κάποιας δύναμης Hx – aL^m στο ανάπτυγμα Taylor. Ας δούμε τα παραδείγματα που ακολουθούν :

```
Normal@seiraD
SeriesCoefficient@seira^2, 4D
```

$$-1 - \frac{1}{2}(-1+x)^2 + \frac{1}{3}(-1+x)^3 - \frac{1}{4}(-1+x)^4 + x$$

$$\frac{11}{12}$$

Teleiώnontaν na poúme óti an h sunάrthsh f **eínai duo metablhtών tóte mporoúme na pároume thn diplή** Series. Έ**tsi h** Series[*f*, 8*x*, x_0, n_x<, 8*y*, y_0, n_y<] **brískei prώta to anάptugma wV proV to** *y*, **kai metά wV proV to** *x*.
P.c

```
Series@Sin@x yD, 8x, 0, 7<, 8y, 0, 7<D
```

$$(y + O(y)^8)\,x + \left(-\frac{y^3}{6} + O(y)^8\right)x^3 + \left(\frac{y^5}{120} + O(y)^8\right)x^5 + \left(-\frac{y^7}{5040} + O(y)^8\right)x^7 + O(x)^8$$

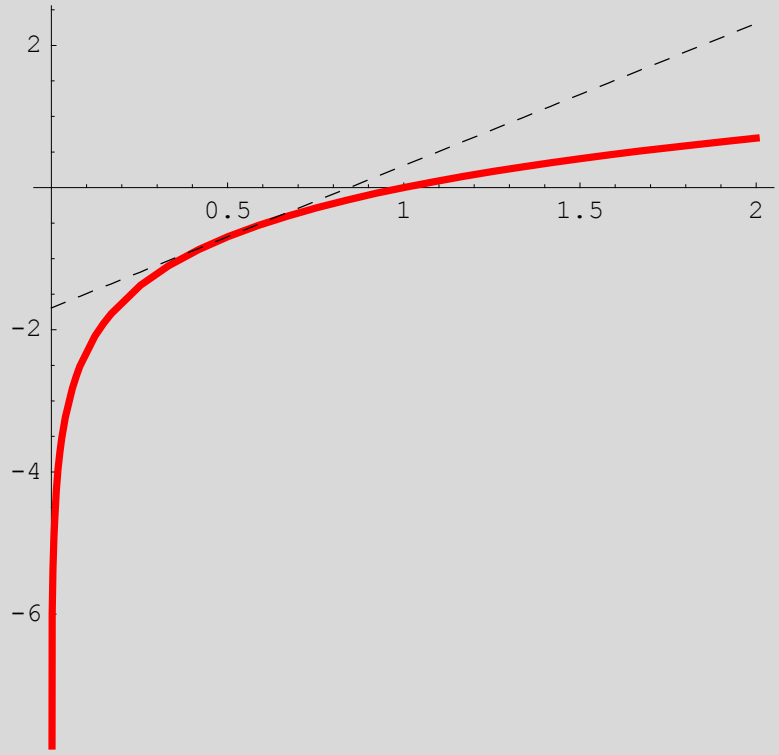
6.6 Hejaptόmenh miaV sunάrthshV

Me thn eukairía twn seirών Taylor **thV** f **gύrw apo to shmeío** a **prépei na poúme óti ta poluwnumikά anaptύgmata pou prokύptoun me thn sunάrthsh** Normal **proseggízoun polύ kalά thn sunάrthsh** f **gύrw apo to** a. ¨**Oso megalύtero to** n **tόso kalύterh h proséggish**. **Ousiastikά loipón ta anaptύgmata** Taylor **apoteloún ta "ejaptόmena" poluώnuma thV** f **sto** a. **Gia** n=1 **écoume thn ejaptόmenh euqeía Parάdeigma**: **gia na broúme thn ejάptomenh sthn** Log[x] **gia a**=0.5 **qétoume** n=1 **sthn** Series **kai metά thn paírnoume thn** Normal

```
efaptomenh = Normal@Series@Log@xD, 8x, .5, 1<DD
```

$$-0.693147 + 2.(-0.5 + x)$$

```
Plot@8Log@xD, efaptomenh<, 8x, 0, 2<,
 AxesOrigin → 80, 0<, AspectRatio → 1, PlotStyle →
  88RGBColor@1, 0, 0D, Thickness@.01D<, 8Dashing@80.02, 0.02<D<<D
```

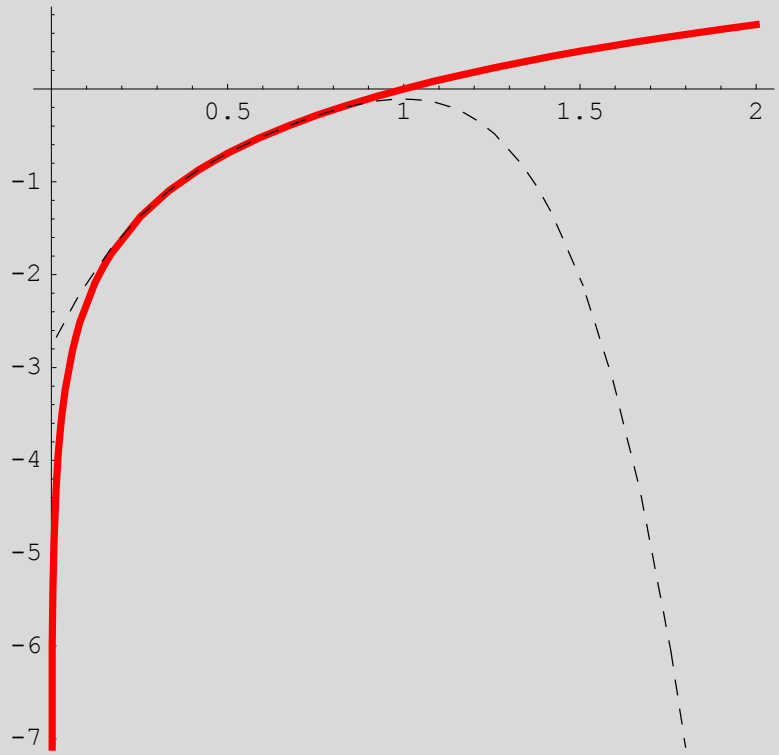



```
¦ Graphics ¦
```

Sto parapánw grájhma scediásame mazí thn Log **kai thn ejaptómenh euqeía. To** RGB[1,0,0] **eínai to kókkino crwma kai to crhsimopoiήsame gia thn** Log **enώ to** Dashing **dhl. tiV diakekomméneV grammέV gia thn ejaptoménh. Parathreíste óti prágmati h euqeía eínai ejaptómenh sto shmeío** a=0.5. **Me crήsh thV** Normal[Series[Log[x],{x,.5,n}]] **gia** n **megalύtera tou** 1 **mporoúme na pároume kalύtereV prosegistikέV kampύleV sthn** Log. **Mporeítai na kánetai thn grajikή parάstash gia** n>1 **gia na deíte thn diajorά. An tώra qéloume mia kalύterh proséggish me polυώnumo p.c** 4 **baqmoύ (gύrw apo to shmeío** a **jusikά)qa prépei na broúme thn** Series 4 **baqmoύ dhladή:**

```
efaptomenh4 = Normal@Series@Log@xD, 8x, .5, 4<DD

-0.693147 + 2. H-0.5 + xL -
 2. H-0.5 + xL2 + 2.66667 H-0.5 + xL3 - 4. H-0.5 + xL4
```

```
Plot@8Log@xD, efaptomenh4<, 8x, 0, 2<,
 AxesOrigin → 80, 0<, AspectRatio → 1, PlotStyle →
  88RGBColor@1, 0, 0D, Thickness@.01D<, 8Dashing@80.02, 0.02<D<<D
```




```
¦ Graphics ¦
```

Άskhsh: **Dínetai h sunάrthsh** f[x_]:=$\grave{e}^{\grave{e}^{\grave{e}^{\grave{e}^{\grave{e}^{\grave{e}^{\grave{e}^{\grave{e}}}}}}}+x}$. **Breíte ta anaptúgmata** Mac-Laurin **thV** f[x] **baqmoú** n=1,2,...20 **me thn boήqeia thV** Series **kai omadopoiήste ta apotelέsmata se mia lísta**. **Sthn sunέceia ejarmόsete thn** Normal **sth lísta kai sthn lísta pou prokúptei qέste** x=0. **Qa prokύyoun** 20 **ariqmoí**. **Exhgeíste giatí autoí oi óroi apotelούn mia suglínousa akolouqía me όrio to** $\grave{e}^{\grave{e}^{\grave{e}}}$. **Póso perípou eínai to** $\grave{e}^{\grave{e}^{\grave{e}}}$ **apo tiV proseggíseiV saV**;

Κεφάλαιο 7ο: Παραγώγιση και ολοκλήρωση.

7.1 Paragώgish kai oloklήrwsh

ΌpwV gnwrízoume h parágwgoV f'[x] dínetai apó ton túpo LimitA$\frac{f[x+h]-f[x]}{h}$, h Ø 0E parádeigma:

```
Remove@fD
f@x_D := 3 x − 2 ê x
LimitA f@x + hD − f@xD
       ---------------- , h → 0E
              h
```

```
3 + 2/x^2
```

Fusiká upárcei kai h sunárthsh D[f[x],{x,n}] pou epistrέjei cwríV kópo thn n-iostή parágwgo wV proV x. Eidiká eán n=1(prώth parágwgoV) tóte mporoúme na gráyoume aplá D[f[x],x] ή pio aplá me f'[x]. Me f''[x] ennooúme thn 2h parágwgo wV proV x k.o.k. An qέloume na broúme thn n-iostή parágwgo cwríV na maV endiajέrei to ónoma thV metablhtήV pou paragwgízoume mporoúme na crhsimopoiήsoume thn
Derivative[n][f]. Sthn Derivative upoqέtoume óti h f έcei móno mia metablhtή(thn #1) kai wV proV autή kai móno paragwgízoume. To apotέlesma qa periέcei autήn thn anώnumh metablhtή.

```
D@f@xD, 8x, 1<D
D@f@xD, xD
f'@xD
Derivative@1D@fD

3 + 2/x^2
```

```
3 + 2/x^2
```

```
3 + 2/x^2
```

```
3 + 2/#1^2 &
```

Gia thn paragώgish sunartήsewn me parapάnw apo mia metablhtέV h diadikasía eínai parómoia.Gia parάdeigma me D[f[x,y],{x,m},{y,n}] **epistrέjetai h merikή parάgwgoV** $\sum^{n+m} f \hat{e} \sum x^m \sum y^n$ **ópou prώta paragwgízoume wV proV** y(n-**iost**ή **parάgwgoV**) **kai metά wV proV** x(m-**iost**ή **parάgwgoV thV prokúptousaV sunάrthshV**). **H** D[f,x,y] **eínai suntomograjía thV** D[f[x,y],{x,1},{y,1}] **dhl**. **thV** $\partial_{x,y} f$. **Apo ta parakάtw paradeígmata eínai janeró óti den prépei na prospaqoúme na paragwgísoume mia sunάrthsh pou écei** If(**dhl**, **o orismόV thV cwrízetai se periptώseiV**) . **Emjanízetai to ídio próblhma pou eícame kai me thn** Limit!

```
Remove@gD
g@x_, y_D := If@y ≠ 0, x Sin@1 ê yD, 0D
D@g@x, yD, 8x, 2<, 8y, 1<D

If@y ≠ 0, 0, 0D
```

```
Remove@gD
g@x_, y_D := x Sin@1 ê yD
D@g@x, yD, 8x, 2<, 8y, 1<D
D@g@x, yD, 8x, 2<D
D@g@x, yD, y, xD

0
```

```
0
```

```
  Cos@ 1/y D
- iiiiiiiiiiiiiiiii
     y2
```

H D[f[x],{x,4}] **par**ά**gei to** ί**dio apot**έ**lesma me to** f''''[x], **me to** D[f[x],x,x,x,x] **kai teloV me to** ∑8x,4< f@xD. **Dial**έ**xte** ό**ti saV ar**έ**sei**! To **s**ύ**mbolo ∑Ñ Ñ to br**ί**skoume sthn pal**έ**tta** BasicInput. **Ed**ώ **pr**έ**pei na pros**έ**coume** ό**tan** έ**coume gin**ό**meno sunart**ή**sewn na b**ά**zoume parenq**έ**seiV alli**ώ**V up**ά**rcei per**ί**ptwsh l**ά**qouV apo dik**ή **maV up**αι**ti**ό**thta p.c an xec**ά**soume tiV parenq**έ**seiV sthn** ∑x Hx Sin@xDL**qa p**ά**roume kata l**ά**qoV**

```
∂x x Sin@xD

Sin@xD
```

dhl.**parag**ώ**gise m**ό**no thn** x **kai** ό**ci to gin**ό**meno**!

Άn **qeloume na paragwg**ί**soume thn** g[y] **wV proV** x **t**ό**te** ό**pwV gnwr**ί**zoume apo thn Qewr**ί**a h par**ά**gwgoV e**ί**nai** 0 **ekt**ό**V kai an qewro**ύ**me** ό**ti h** y **den e**ί**nai mia staqer**ά **all**ά **mia sun**ά**rthsh tou** x. **Aut**ό **dhl**ώ**netai me thn sun**ά**rthsh** NonConstants **p.c**

```
Remove@gD
D@g@yD, x, NonConstants → yD
Remove@g, fD
D@f@g@tDD, x, NonConstants → tD

D@y, x, NonConstants → 8y<D gÅ@yD
```

```
D@t, x, NonConstants → 8t<D fÅ@g@tDD gÅ@tD
```

Sto deύ**tero par**ά**deigma qewr**ή**same** ό**ti h metablht**ή t **einai mia sun**ά**rthsh thV** x.

7.2 Εύρεση τοπικών ακρότατων με χρήση των μερικών παραγώγων

Έχουμε ήδη αναλύσει την συνάρτηση FindMinimum για την εύρεση κάποιων ακροτάτων μιας συνάρτησης f. Όμως εδώ θα προσπαθήσουμε να βρούμε τα τοπικά ακρότατα ή τουλάχιστον τα "πιθανά" τοπικά ακρότατα της f λύνοντας την εξίσωση f'[x]==0 αν η f έχει μόνο μια μεταβλητή ή ένα σύστημα εξισώσεων αν έχει παραπάνω (π.χ $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$ αν η f έχει δύο μεταβλητές). Για την λύση των συστημάτων αυτών μπορούμε να χρησιμοποιήσουμε την Solve. Αν οι παράγωγοι δεν είναι πολυώνυμα τότε προτιμάμε να βρούμε αριθμητικές λύσεις με την NSolve ή την FindRoot. Εδώ θα δούμε ότι η FindRoot έχει την μορφή FindRoot[$\{eqn_1, eqn_2, \ldots\}, \{x, x_0\}, \{y, y_0\}, \ldots$]όπου $eqn_1, eqn_2, \ldots$είναι το σύστημα των εξισώσεων και $x_0, y_0, \ldots$είναι κάποιοι αριθμοί "κοντά" σε κάποια πραγματική λύση του συστήματος. Αυτό είναι και το μεγάλο πρόβλημα με την FindRoot: πρέπει να γνωρίζουμε κοντά αλλιώς μπορεί να μην βρεί καμμία λύση! Παραδείγματα: Αν προσπαθήσουμε να βρούμε τα τοπικά ακρότατα της x Sin[1/y] με την NSolve και την FindRoot και για x=0.05

```
g@x_, y_D := x  Sin@1 ê yD
g1@x_, y_D := D@x  Sin@1 ê yD, xD
g2@x_, y_D := D@x  Sin@1 ê yD, yD
g2@x, yD
z = NSolve@g2@0.05, yD « 0, yD
0.05 Sin@1 ê yD ê. z
```

$$-\frac{x \cos\left(\frac{1}{y}\right)}{y^2}$$

```
Solve::ifun :
 Inverse functions are being used by Solve, so some solutions may not be found.
```

```
88y → −0.63662<, 8y → 0.63662<<
```

```
8−0.05, 0.05<
```

```
FindRoot@g2@0.05, yD « 0, 8y, 0.1, −1, 1<D
```

```
General::ivar : 0.1` is not a valid variable.
```

```
General::ivar : 0.1` is not a valid variable.
```

```
FindRoot::frnum :
 Function 8∂0.1H−0.0272011L< is not a length 1 list of numbers at 8y< = 80.1<.
```

```
FindRoot@g2@0.05, yD == 0, 8y, 0.1, −1, 1<D
```

Na exhgήsoume óti h NSolve **ébgale polú eúkola éna topikó akrótato sto shmeío** y=0.63662(**kai állo éna sto** y**Ø**-0.63662) . **Eínai mégisto dióti sto prohgoúmeno kejálaio eícame kánei kai thn grajikή parástash thV** g2[0.05,y]. **Dustucwv den ta brήke óla**. **Me thn** FindRoot **mporoúme na broúme kai álla**! **To problhma eínai pwV dialégoume to arcikó maV** y0 **gia na xekinήsei to yáximo**. **Qa prépei na kánoume diájoreV dokiméV sto** y0 **kai gia diájora diastήmata**. **Állo éna mikrótero próblhma**: **prépei na gráyoume ton túpo thV** g2 **apeuqeíaV** (**cwríV crήsh tou orismoú tou**) **alliώV bgaínoun perίerga mhnúmata ópwV t**o `FindRoot::frnum : Function 8∂0.1H−0.0272011L<is not a length 1 list of numbers at {y} = {0.1}`.

```
Remove@g2D
                  x CosA1yE
g2@x_, y_D := −  iiiiiiiiiiiiiiiiiiii
                      y2
ymin = −1
ymax = 1
z = FindRoot@g2@0.05, yD == 0, 8y, 0.1, −1, 1<D
g@0.05, yD ê. z
```

```
−1
```

```
1
```

```
8y → 0.0909457<
```

```
−0.05
```

Epanalambánoume xaná thn genikή arcή: ó**tan orízoume sunartήseiV**: **na mhn crhsimopoioúme álleV ήdh kataskeuasméneV allá na tiV orízoume "kateuqeían"**. Étsi **den qa ήtan kalό gia parádeigma**, **na orízame** g[x,y]:=x*Sin[1/y] **kai sthn sunéceia** g2[x_,y_]:=D[g[x,y],y] **allá qa ήtan protimότero na orízame kat'euqeían** g2[x,y]:=− $\frac{x \cos\left[\frac{1}{y}\right]}{y^2}$. **An qéloume perissότereV lúseiV me megalύterh akríbeia qa prépei na alláxoume thn arcikή timή tou** y0 **na qésoume mikrότera diastήmata** ymin, ymax **kai na bάloume epiplέon orísmata mésa sthn** FindRoot **p.c**

```
z = FindRoot@g2@0.05, yD == 0, 8y, 0.01, 0.001, 0.09<,
  AccuracyGoal → 24, WorkingPrecision → 34,
    MaxIterations → 50D
g@0.05, yD ê. z
```

```
8y → 0.010105075751866370525008492912540 59<
```

```
−0.05
```

Άskhsh: Dínetai h sunárthsh f[x,y]:=x^2+ x*y+y^2-2*x-6*y. **Na breíte ta kritiká shmeía thV dhl**. **ta shmeía pou mhdenízontai oi merikéV parágwgoi**. **Sthn sunéceia crhsimopoiώntaV thn** FindMinimum **prospaqήsete na apantήsete an ta akrótata pou brήkate eínai topiká mégista** ή **topiká elάcista**.**TéloV crhsimopoieíste thn** ContourPlot **gia thn** f **kai gia katάllhla diastήmata** {xmin,xmax} **kai** {ymin,ymax}**stouV άxoneV** Ox **kai** Oy **antístoica** (**pou na periécoun to** ή **ta shmeía pou brήkate**) **étsi ώste na deíte an prágmati h grajikή parάstash pou paírnetai sumjwneí me ta apotelésmata pou bgάlate**.(**Prosocή prin kalésetai thn** ContourPlot[x^2+x*y+y^2-2*x--6*y,{x,xmin,xmax},{y,ymin,ymax}] **na gráyete** <<Graphics`Graphics3D` **dióti káqe sunárthsh gia grajikéV parastάseiV écei to dikó thV pakéto ektóV apó merikéV pou den creiάzetai na kalésoume to pakéto touV sióti jortώnetai móliV anoíxoume to** *Mathematica*). **Me to** Help **prospaqήste na máqete perissótera gia thn** ContourPlot **ώste na kánete to grájhma saV elkustikó**.(**p.c dώste crώma bάzontaV mésa sthn** ContourPlot **to** ColorFunction**Ø**Hue. **Sto ídio pakéto anήkei kai h** ShadowPlot3D **epíshV me polú wraía apotelésmata**). **H** Plot3D(**autή den creiάzetai kanéna idiaítero pakéto**) **sthn perίptwsh maV den bohqά polú, ektóV kai an allάxete lígo ton crwmatismó thV epijáneiaV pou prokúptei me to epiplέon carakthristikó** `ColorFunction → Hue` **mesa sthn** Plot3D.

7.3 Aórista oloklhrώmata

H basikή entolή gia na broúme to aóristo oloklήrwma wV proV x **eínai h** Integrate[f[x],x]. **p.c**

```
Integrate@f'@xD, xD
D@Integrate@f@xD, xD, xD

f@xD
```

```
f@xD
```

Dhladή to oloklήrwma thV paragώgou thV f **wV proV** x **eínai h ídia h** f! **Kai h parágwgoV tou oloklhrώmatoV thV** f **wV proV** x **eínai páli h** f. **Me άlla lógia h oloklήrwsh me thn paragώgish eínai antístrojeV sunartήseiV**.**AV kánoume kai άlla paradeígmata**:

```
Integrate@Log@xD, xD
Integrate@x^2 Cos@n xD, xD
```

```
‡ x^n ăx
```

```
Integrate@Log@x yD, x, yD
Integrate@Cos@Sin@xDD, xD
```

```
‡ E^(1-x²) ăx
```

```
-x + x Log@xD
```

$$\frac{2\,x\,\text{Cos}[n\,x]}{n^2}+\frac{(-2+n^2\,x^2)\,\text{Sin}[n\,x]}{n^3}$$

$$\frac{x^{1+n}}{1+n}$$

```
-2 x y + x y Log@x yD
```

```
‡ Cos@Sin@xDD ăx
```

$$\frac{1}{2}\sqrt{\pi}\,\text{Erf}[x]$$

AV arcísoume na kánoume parathrήseiV:

1)**Sto aóristo oloklήrwma paraleípetai h staqerá sto téloV**!

2)**Mporoúme na oloklhrώsoume akóma kai an h sunárthsh maV écei mia** ή **parapánw paramétrouV** (**pc thn** n).

3) **Ta aórista oloklhrώmata upologízontai me thn siwphrή upóqesh óti oi crhsimopoioúmeneV parámetroi eínai tétoieV ώste óso h oloklhrwtéa sunárthsh óso kai to apotélesma na écoun énnoia**. Étsi, **gia parádeigma**, **to oloklήrwma** $\frac{x^{1+n}}{1+n}$ **pou epistréjei to** *Mathematica* **gia thn oloklήrwsh** ÿ x^n ăx **xéroume óti eínai alhqέV**, **upó thn proupóqesh óti to** n **eínai diajoretikó tou** -1, **ajoú**, **wV gnwstó**, ÿ x^{-1} **„** x=ln[x].

4)To **sύmbolo thV oloklήrwshV** ÿ **Ñ „ Ñmporoúme na to broúme apo tiV paléteV** 3BasicInput. ÉnaV **állοV trópoV cwríV paléteV eínai na patήsoume diájorouV sunduasmoύV plήktrwn p.c gia na gráyoume to orisméno** $\int_0^7 f[x]\,dx$ **qa prépei na patήsoume**: int + 0 % 7 f[x] dd x (**ta** [**kai**] **den ta ktupáme**! **to** +]**gia parádeigma shmaínei óti patáme mazí to kai to** + **enώ to eínai to mazí me to** SPACE **plήktro**) .

5)**Mporoúme na écoume kai diplά** (**allά kai pollaplά oloklhrώmata**) ópwV **gia parádeigma to** Integrate[Log[x y],x,y] **pou isodunamei me** ÿ dx ÿ dy Log[x, y]**dhl**. **prώta oloklhrώnoume wV proV** y **kai to apotélesma metά wV proV** x.

6) **Upárcei éna megálo plhqoV oloklhrwmάtwn**, **ta opoía upologízetai me thn crήsh eidikών**

sunartήsewn ópwV **oi** Erf, EulerGamma, Fresnel, Hypergeometric, Elliptic **kai** álleV. **Mporeítai na breíte plhrojoríeV sto** help.

7) **Upárcoun oloklhrώmata ta opoía den mporeí na upologísei to** *Mathematica*. ***Se mia*** **tétoia perίptwsh epistrέjetai to ídio to oloklήrwma p.c sto** ÿ Cos@Sin@xDD ăx. **Fusiká án antí aóristo écoume orismέno oloklήrwma tóte mporoúme na pároume mia ariqmhtikή timή tou me thn crήsh thV** N **ή thV NI**ntegrate ópwV **qa doúme parakátw. p.c**

```
olokl = NIntegrate@Cos@Sin@xDD, 8x, 0, Pi ê 3<D

0.89975
```

7.4 Orismέna oloklhrώmata

H basikή entolή gia ta orismέna oloklhrώmata eínai ópwV **kai sta aórista me thn eisagwgή twn oríwn p.c t**o **orismέno oloklήrwma** $\int_0^{Pi/3} Sin[x]^2 Cos[x]^3 \, dx$ **apl**á **qa to gráyoume**

```
Integrate@Sin@xD^2 Cos@xD^3, 8x, 0, Pi ê 3<D

11
ÄÄÄÄ
160
```

Mporoúme jusiká na crhsimopoiήsoume thn 3BasicInput **paléta ή patώntaV plήktra.Mporoúme jusiká na báloume kai to ápeiro se éna ή kai sta dύo ákra**

```
Integrate@Sin@xD^2 Cos@xD^3, 8x, 0, ∞<D

Integrate::idiv : Integral of Cos@xD^3 Sin@xD^2 does not converge on 80, ∞<.

∞
‡ Cos@xD^3 Sin@xD^2 ăx
0
```

Se autήn thn perίptwsh to oloklήrwma den upárcei! Se állh perίptwsh ómwV upárcei p.c

```
IntegrateAE^-x^2, 8x, 0, ∞<E

Π
ÄÄÄ
2
```

EpίshV se álleV periptώseiV to *Mathematica* **mporeί na maV apantήsei me to** If **p.c**

```
Integrate@x^n, 8x, 0, 1<D

                1
IfARe@nD > -1, ÄÄÄÄÄÄ, ‡ x^n ăxE
               1 + n
                        0
```

Autό shmaínei óti eán to pragmatikό méroV thV paramέtrou n (**dióti mporeί kápoioV na dώsei kai migadikή timή sto** n) **eínai** > **tou** -1 **tóte to oloklήrwma eína ίso me** $\frac{1}{1+n}$ **alliώV den mporeί na dώsei apάnthsh! P.c aV báloume** n=2

```
% ê. n → 2
```

$$\frac{1}{3}$$

Genikά an epiqumóume na qésoume kάpoiouV periorismoύV ston upologismó tou orisménou oloklhrώmatoV qa prépei na tiV eisάgoume me thn entolή qa Assumptions-> **oi periorismoί p.c an qeloume na dhlώsoume óti mia parάmetroV** m **paírnei pragmatikέV timέV (kai óci migadikέV) qa mporoύsame na to dhlώsoume légontaV óti to jantastikó méroV eínai** 0 : Im[m]==0 **p.c**

```
Integrate@Sin@m xD ê x, 8x, 0, Infinity<, Assumptions → Im@mD « 0D
```

$$\frac{1}{2}\,\pi\,\text{Sign}[m]$$

An den bάzame ton periorismó autó qa paírname:

```
Integrate@Sin@m xD ê x, 8x, 0, Infinity<D
```

$$\text{If}\left[\text{Im}[m] == 0,\ \frac{1}{2}\,\pi\,\text{Sign}[m],\ \int_0^{\infty} \frac{\text{Sin}[m\,x]}{x}\,dx\right]$$

H entolή NIntegrate[f[x],{x,a,b}] **epistréjei mia ariqmhtikή prosέggish tou orisménou oloklhrώmatoV** $\int_a^b f[x]\,dx$ **enώ upάrcoun kai crήsima epiplέon carakthristikά pou mporoύme na prosqέsoume όpwV gia parάdeigma** AccuracyGoal-> 20 **gia na beltiώsoume thn akríbeia twn upologismwn**. **P.c**

```
NIntegrate@Sin@xD^2 Cos@xD^3, 8x, 0, Pi ê 3<,
 AccuracyGoal → 20, WorkingPrecision → 30D
```

```
0.11907849302036031393
```

H sunάrthsh NIntegrate **eínai crήsimh ótan h** Integrate **den mporeί an maV bgάlei éna apotέlesma**. **H** Integrate **kάnei sumbolikoύV upologismoύV enώ h** NIntegrate **crhsimopoieί proseggistikέV ariqmhtikέV meqódouV**.**P.c**

```
H Integrate@Abs@x-Log@xD+1.5D,8x,1,3<D L
NIntegrate@Abs@x - Log@xD + 1.5D, 8x, 1, 3<D
```

```
5.70416
```

Bálame se scólia to prώto oloklήrwma dióti se maV to *Mathematica* **den mpórese na bgálei kάpoio apotélesma parólou pou periméname arketή ώra**! **ΌmwV polύ grήgora h** NIntegrate **upológise to oloklήrwma**!

7.5 Pollaplά oloklhrώmata

Me thn entolή Integrate **mporoύme na upologίsoume kai diplά allά kai triplά k.o.k oloklhrώmata arkeί na diamorjώsoume katάllhla ta όria** D **tou oloklhrώmatoV**. **AV doύme merikέV periptώseiV**
1)**H periocή** D **apoteleίtai apo kάpoia diastήmata p.c** $D = \{x, y\} \in \mathbb{R}^2 \text{ êa } x \le b,\ c \le y \le d\}$. **Tóte upologίzoume ta oloklhrώmata qétontaV ta katάllhla όria** {x,,a,b}, {y,c,d} **p.c**

```
Integrate@x y^2 z^3 , 8x, 0, 1<, 8y, 0, 1<, 8z, 0, 2<D
```

$$\frac{2}{3}$$

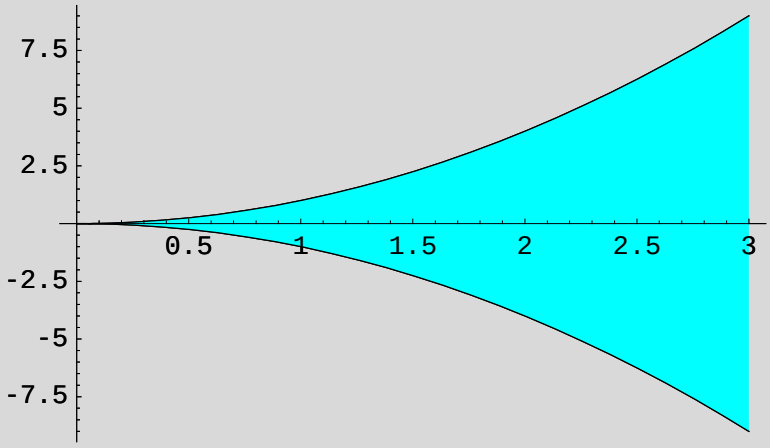
Se autήn thn perίptwsh den pάizei rόlo h seirά oloklήrwshV:

```
Integrate@x y^2 z^3 , 8z, 0, 2<, 8x, 0, 1<, 8y, 0, 1<D
```

$$\frac{2}{3}$$

2) **H periocή** D **eínai mia periocή tou didiάstatou cώrou pou perikleίtai apo kάpoieV kampύleV** ή **mia periocή tou trsdiάstatou cώrou pou perikleίetai apo kάpoieV epijάneieV k.o.k tóte qa prépei na prosέxoume thn seirά pou oloklhrώnoume**(**dhl**. **wV proV poiά metablhtή qa oloklhrώsoume prώta**, **metά poia akolouqeί k.o.k**). **Akolouqoύn merikά paradeίgmata** D **kai sunartήsewn** f **kai h seirά pou oloklhrώnoume**. **Gia na gínoun kai pio parastatikά qa kάnoume prώta thn grajikή parάstash twn cwrίwn** D. **Sthn didiάstath perίptwsh twn** D **qa creiasteί na kalέsoume to pakéto** Graphics`Filled-Plot`

```
<< Graphics`FilledPlot`
FilledPlot@8x^2, -x^2<, 8x, 0, 3<D
```




```
¦ Graphics ¦
```

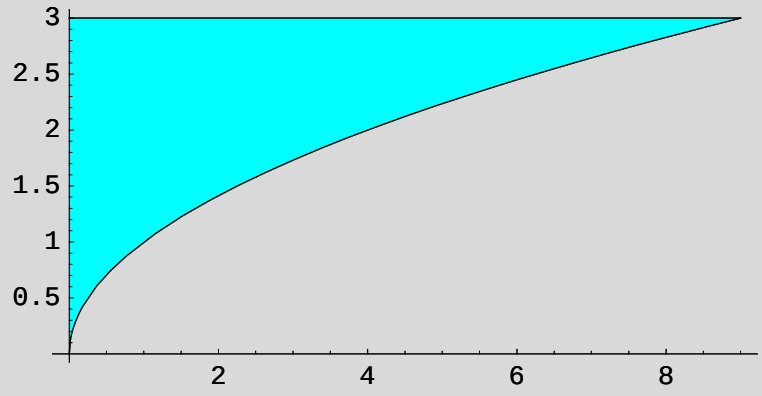
edώ D **=** 8Hx, yL **œ** Ñ² ê0**§**x**§**3, -x^2 **§** y**§**x^2} **kai oloklhrώnoume mia** f **p.c thn** 2 x -3 y^2 **prώta wV proV** y **kai metά wV proV** x **dhl**.

```
Integrate@2 x - 3 y^2, 8x, 0, 3<, 8y, -x^2, x^2<D
```

$$-\frac{3807}{7}$$

Prέpei na prosέxoume óti to oloklήrwma pou prέpei na upologisteί prώto, **mpaίnei pάntote sto téloV tou** Integrate! Άllo **par**άdeigma:

```
<< Graphics`FilledPlot`
FilledPlot@8Sqrt@xD, 3<, 8x, 0, 9<,
 AxesOrigin → 80, 0<, AspectRatio → 1 ê 2D
```




```
¦ Graphics ¦
```

edώ $D = \{(x, y) \in \mathbb{R}^2 : 0 \le y \le 3,\ 0 \le x \le y^2\}$ **kai oloklhrώnoume prώta wV proV** x **kai metά wV proV** y **dhl**.

```
Integrate@2 x - 3 y^2, 8y, 0, 3<, 8x, 0, y^2<D
```

$$-\frac{486}{5}$$

Genikά na écoume sto noú óti oloklhrώnoume teleutaía se diάsthma pou écei staqerά άkra dhl. **eínai thV morjήV tou kleistoú diastήmatoV** [**a**,**b**]. **p.c an écoume thn** f[x,y,z]:=x y^2 z^3 **kai qéloume na broúme to triplό oloklήrwma sto cwrío** $D = \{(x, y, z) \in \mathbb{R}^3 : 0 \le x \le 1, 0 \le y \le 1-x,\ 0 \le z \le x*y\}$ **pou perikleíetai apo ta epípeda** x+y=1 **kai** z=0, **kai thn epijάneia(uperbolikό paraboloeidéV)** z=x*y . **Lógw thV morjήV tou** D **oloklhrώnoume prώta wV proV** z **metά wV proV** y **kai téloV wV proV** x.

```
Integrate@x y^2 z^3, 8x, 0, 1<, 8y, 0, 1 - x<, 8z, 0, x y<D
```

$$\frac{1}{288288}$$

Bébaia to ídio oloklήrwma qa mporoúsame na to kάnoume diadocikά se tría bήmata allά jusikά eínai kourastikό...

```
int1 = Integrate@x y^2 z^3, 8z, 0, x y<D
int2 = Integrate@int1, 8y, 0, 1 - x<D
int3 = Integrate@int2, 8x, 0, 1<D
```

$$\frac{x^5\, y^6}{4}$$

$$\frac{1}{28}\, H1 - xL^7\, x^5$$

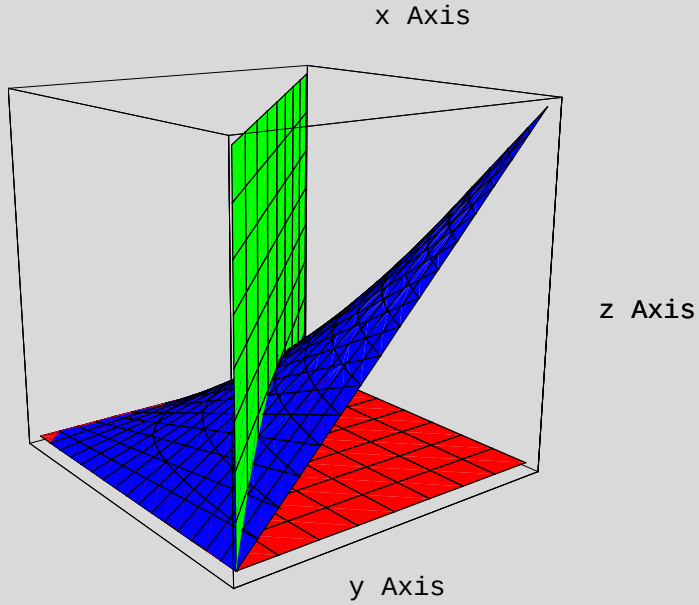
$$\frac{1}{288288}$$

Parakάtw qa prospaqήsoume na scediάsoume to D. **Gia boήqeia pάnw sth grajikή sunάrthsh** ContourPlot3D **pou crhsimopoioύme mporeíte na anatréxete sto** Help **kai epíshV na "paíxete" me tiV grajikέV parastάseiV ajairώntaV ή allάsontaV kάpoia carakthristikά ώste na katalάbete ti akribώV kάnoun**. **Qa poύme monάca óti h** plot1 **paristάnei to epípedo** z==0 **me kόkkino crώma**(**Scólio**: **allάxame to** z==0 **se kάti scedón όmoio** z-0.00001==0 **dióti den mporέsame na crwmatisoume to epípedo** z==0 **me thn** ContourPlot3D!!) **h** plot2 **paristάnei to epípedo** x+y=1 **se prάsino kai h** plot3 **to paraboloeidέV** z=x y **me mplέ crώma**. To D **eínai to kleistó cwrío pou perikleíetai metaxú twn triών epijaneiών**.

```
<< Graphics`ContourPlot3D`
plot1 = ContourPlot3D@z - 0.00001, 8x, 0, 1<, 8y, 0, 1<, 8z, 0, 1<,
  ContourStyle → RGBColor@1, 0, 0D, DisplayFunction → IdentityD
plot2 = ContourPlot3D@x + y - 1, 8x, 0, 1<, 8y, 0, 1<, 8z, 0, 1<,
  ContourStyle → RGBColor@0, 1, 0D, DisplayFunction → IdentityD
plot3 = ContourPlot3D@z - Hx yL, 8x, 0, 1<,
  8y, 0, 1<, 8z, 0, 1<, PlotPoints → 85, 5<,
  ContourStyle → RGBColor@0, 0, 1D, DisplayFunction → IdentityD
Show@plot1, plot2, plot3, Axes → True, Ticks → 8None, None, None<,
 AxesLabel → 8"x Axis", "y Axis", "z Axis"<,
 Lighting → False, DisplayFunction -> $DisplayFunction,
 ViewPoint -> 82.434, -1.853, 0.866<D


¦ Graphics3D ¦
```

```
¦ Graphics3D ¦
```

```
¦ Graphics3D ¦
```




```
¦ Graphics3D ¦
```

Antí tou parapánw plot1 **mporeíte kai na bálete éna kókkino polúgwno wV exήV**

```
plot1 = Graphics3D@
  8RGBColor@1, 0, 0D, Polygon@880, 0, 0<, 81, 0, 0<, 81, 1, 0<, 80, 1, 0<<D<,
  DisplayFunction → IdentityD
```

dokimάste parapάnw na deíte thn diajorά!

Άskhsh: Na upologisteí to diplό oloklήrwma thV $f[x, y] = \frac{1}{E^x \operatorname{Sqrt}[y]}$ **sthn periocή** $D = \{\{x, y\} \in \mathbb{R}^2 \mid -2 \le y \le 0, |x-1| \le 4\}$ **kai na gínei grajikή parάstash tou** D.

Κεφάλαιο 8ο: Διαφορικές εξισώσεις

8.1 Η entolή DSolve

Se autó to kejálaio qa ascolhqoúme me thn lúsh diajorikón exisώsewn dhl. exisώsewn me paragώgouV stiV opoíeV eínai ágnwsth mia ή perissótereV sunartήseiV y[x] **kai pou h parágwgoV touV** y'[x] **ikanopoieí tiV exisώseiV. P.c** x^2 y'[x]-y[x]==0 (**ágnwsth h** y[x]) **kai to sύsthma** y'[x]+y[x]-z[x]==Sin[x],z'[x]+y[x]+z[x]==Cos[x] **me ágnwsteV sunartήseiV tiV** y[x],z[x]. **Duo eínai oi basikéV sunartήseiV h** DSolve **kai h** NDSolve. H **deúterh maV parécei thn dunatóthta ariqmhtikήV epílushV diajorikών exisώsewn kai susthmátwn kai crhsimopoieí proseggistikéV meqódouV enώ h prώth prospaqeí na breí mia akribή lúsh twn exisώsewn**.

```
?DSolve
```

```
DSolve@eqn, y, xD solves a differential equation for
   the function y, with independent variable x. DSolve@
   8eqn1, eqn2, ...  <, 8y1, y2, ...  <, xD solves a
   list of differential equations. DSolve@eqn, y, 8x1,
   x2, ... <D solves a partial differential equation. More...
```

dhl. h DSolve[eqn, y, x] **lúnei thn diajorikή exíswsh** eqn **kai brískei thn sunárthsh** y, **h opoía écei anexárthth metablhtή thn** x. **H** DSolve[{eqn1, eqn2, ... }, {y1, y2, ... }, x] **lúnei to sύsthma** eqn1, eqn2, ... **apo diajorikéV exisώseiV me ágnwsteV tiV sunartήseiV** y1, y2, ... **pou eínai sunartήseiV thV** x. **TélοV h** DSolve[eqn, y, {x1, x2, ... }] **lúnei thn exíswsh merikών paragώgwn** eqn **wV proV thn** y **pou eínai sunárthsh twn metablhtών** x_1, x_2, ... **Akolouqoún arketá paradeígmata (gia didaktikoúV lógouV gráyame katá "lάqoV"** DSolve[eqn,y[x],x] **antí tou swstoú** DSolve[eqn,y,x])

```
Remove@eqn, y, xD
eqn = x^2 y'@xD - y@xD « 0;
DSolve@eqn, y@xD, xD

88y@xD → Ą^-1êx C@1D<<
```

Blépoume óti h genikή lúsh periécei mia staqerά C[1]. **An perilambánontai kai álleV staqeréV sthn lúsh autéV qa apariqmoúntai wV** C[1],[2],....**Mporoúme na alláxoume ton sumbolismó twn staqerών ópwV maV arései p.c to** C **me** sta8era **me thn entolή** DSolveConstants->sta8era **p.c**

```
soll = DSolve@eqn, y@xD, x, DSolveConstants → sta8eraD

88y@xD → Ą^-1êx sta8era@1D<<
```

Prosέxte óti h lúsh y[x] **den dínetai apeuqeíaV allά mésa se** {{}}!!! **dhl**. **mésa se dúo lísteV ópwV jaínetai kai apo thn** FullForm:

```
FullForm@sollD

List@List@Rule@y@xD,
   Times@Power@E, Times@-1, Power@x, -1DDD, sta8era@1DDDDD
```

PwV ómwV qa mporέsome na "xetrupώsoume" thn y[x]] **apo kei mésa gia na broúme gia parάdeigma to tetrάgwnó thV**; **Qa anajéroume dúo trópouV**:
PrώtoV trópoV crhsimopoiώntaV thn Last ή **isodúnama paírnontaV to stoiceío** 2o **thV** soll[[1,1]] **dhl**. **to** soll[[1,1]][[2]] **pou grάjetai aploústera** soll[[1,1,2]]

```
soll@@1, 1, 2DD
Last@soll@@1, 1DDD

Ą^-1êx sta8era@1D
```

```
Ą^-1êx sta8era@1D
```

```
Ą^-1êx sta8era@1D
```

DeύteroV trópoV: **crhsimopoiώntaV to sύmbolo** /. **thV antikatάstashV**:

expr **/.** *rules* **ejarmózei touV kanόneV pou brískontai sta** rules **gia na metaschmatísei kάqe tmήma thV** *expr*.

```
y@xD ê. soll

8Ą^-1êx sta8era@1D<
```

Epeidή ómwV upάrcoun pάli ta {} **qa prépei kalύtera na grάyoume**

```
y@xD ê. soll@@1, 1DD

Ą^-1êx sta8era@1D
```

ή **isodύnama**

```
y@xD ê. soll@@1DD


Ą^-1êx sta8era@1D
```

(To y[x]/.soll[[1]] **aplά shmaίnei óti brískoume prώta to** y[x]/.soll **dhl**. **th lίsta** 8Ą^-1êx sta8era@1D< **kai metά brískoume to** 1**o stoiceίo thV lίstaV dhl**. **to** Ą^-1êx sta8era@1D(=y[x]/.soll[[1]]).)
Tώra apo thn stigmή pou écoume ton túpo thV y[x] **mporούme na kάnoume dokimή an dhl**. **prάgmati ikanopoieίtai h arcikή maV diajorikή exίswsh**:

```
lysh = soll@@1, 1DD
eqn ê. lysh

y@xD → Ą^-1êx sta8era@1D
```

```
-Ą^-1êx sta8era@1D + x^2 y'@xD == 0
```

DustucώV den apέdwse h antikatάstash y@xD → Ą^-1êx sta8era@1D**mésa sthn** eqn(**den pήrame** True) .**Autó sumbaίnei dióti gίnetai antikatάstash thV sunάrthshV** y[x] **allά óci kai thV paragώgou thV** y'[x] **mésa sthn** eqn!!! **Me** y@xD → Ą^-1êx sta8era@1D**ennooύme antikatάstash tou** y[x] **me** Ą^-1êx sta8era@1D**kai típota parapάnw**. **Me authn thn antikatάstash den mporeί na katalάbei to** Mathematica **óti me** y[x] **ennooύme sunάrthsh tou** x !!! **Opóte den écei kammiά idéa gia to pwV qa breί mia timή thV** y **ή pwV qa paragwgίsei thn** y. **To móno pou xέrei eίnai h timή thV** y[ ] **ótan écoume qέsei to grάmma** x. **Opóte qa prépei prώta na kάnoume thn antikάtάstash thV** y[x]**me** y[x]/.soll[[1,1]] **kai tou** x **me** 3, **gia na upologίsoume gia parάdeigma to** y[3] **kai prépei na broύme thn parάgwgo** y1 **thV** y **kai metά na antikatastήsoume sthn** eqn **sugcrónwV tiV antikatastrάtieV twn** y[x],y'[x] **gia na kάnoume thn dokimή**. **Thn parάgwgo** (**se morjή kanóna**) **thn brískoume me** D[soll,x][[1,1]] **kai se morjή sunάrthshV me** D[y[x]/.soll[[1,1]],x]. **AV doύme ta parakάtw apotelέsmata**

```
y@3D = y@xD ê. soll@@1DD ê. x → 3
paragwgoslyshs = D@soll, xD@@1, 1DD
y1@xD = D@Hy@xD ê. soll@@1, 1DDL, xD
eqn ê. 8lysh, y'@xD → y1@xD<
eqn ê. 8lysh, paragwgoslyshs<
```

```
sta8era@1D
Ą1êš
```

```
y'@xD → A^-1êx sta8era@1D
                 x²
```

```
A^-1êx sta8era@1D
         x²
```

```
True
```

```
True
```

To /.{lysh, y'[x]**Ø**y1[x]} **shma**í**nei** ó**ti sugcr**ó**nwV antikaqisto**ú**me thn** y[x] **me** Ą^-1êx sta8era@1D **kai thn** y'[x] **me** y1[x] **mé**s**a sthn** eqn. **AV k**á**noume kai** é**na apl**ó **par**á**deigma gia na katal**á**boume thn diajor**á:

```
8a, b, c< ê. a → b ê. b → d

8d, d, c<
```

```
8a, b, c< ê. 8a → b, b → d<

8b, d, c<
```

Sto 1**o par**á**deigma antikatast**ή**same diadocik**á enώ **sto** 2**o k**á**name mia antikat**á**stash sugcr**ó**nwV**! **Giaut**ó exá**llou pro**é**kuyan kai diajoretik**á **apotel**é**smata**!
Gia na katalá**bei to** *Mathematica* ó**ti me** y[x] **ennoo**ú**me mia sun**á**rthsh tou x qa mporo**ú**same gia par**á**deigma na or**í**soume sth q**é**sh thV** y **thn** f[z_]:=y[x]/.soll[[1]]/.x**Ø**z **kai sth sun**é**ceia** ó**pou up**á**rcei h** y **na th antikaqisto**ú**me me thn** f :

```
f@z_D := y@xD ê. soll@@1DD ê. x → z
f'@zD
eqn ê. y → f

A⁻¹êz sta8era@1D
      Z²
```

```
True
```

Scólio: An antí y[x] **qésoume aplá** y **mésa sthn** DSolve **tóte ta prágmata aplousteúontai**. **Prosέxte prosektiká ta apotelésmata parakátw**:

```
y@xD ê. soll@@1, 1DD
D@y@xD ê. soll@@1, 1DD, xD
y@zD ê. soll@@1, 1DD
D@y@zD ê. soll@@1, 1DD, zD
Clear@yD
soll0 = DSolve@eqn, y, xD
y@3D ê. soll0@@1, 1DD
y@zD ê. soll0@@1, 1DD
D@y@zD ê. soll0@@1, 1DD, zD


Ą^-1êx sta8era@1D
```

```
Ą^-1êx sta8era@1D
          x²
```

```
y@zD
```

```
y^Ì@zD
```

```
88y → Function@8x<, Ą^-1êx C@1DD<<
```

```
C@1D
Ą^1ê3
```

```
Ą^-1êz C@1D
```

```
Ą^-1êz C@1D
          z²
```

prosocή ά**llo apot**é**lesma me thn** DSolve[eqn,y,x] **kai** ά**llo** DSolve[eqn,y[x],x]. **H** DSolve[eqn,y,x] **e**í**nai pol**ύ **cr**ή**simh di**ό**ti ma**V **d**í**nei to** y **na e**í**nai sun**ά**rthsh tou** x **kai o t**ύ**poV thV** y **e**í**nai** y[x]/soll0[[1,1]] **dhl**. =Ą^-1êx C@1D. ¨**Etsi gia par**ά**deigma to** y[z] **e**í**nai** í**so me** y[z]/.soll0[[1,1]] **dhl**. Ą^-1êz C@1D**kai h par**ά**gwgoV wV proV** z **e**í**nai** D[y[z]/.soll0[[1,1]], z] **dhl**. Ą^-1êz C@1D . **Parathre**í**ste** ό**ti to** *Mathematica* **gia thn ant**í**stoich ap**ά**nthsh pou d**í**nei h** DSolve[eqn,y[x],x] , **mpore**í **na bre**í **thn par**ά**gwgo wV proV** x **all**ά **den gnwr**í**zei pwV na brei to** y[z] **o**ύ**te thn par**ά**gwgo wV proV** z(ό**pwV** ή**dh proanaj**é**rame**)!!!

To pleoné**kthma e**í**nai janer**ό: **mporo**ύ**me me thn l**ύ**sh gia thn** ά**gnwsth** y **pou d**í**nei h** DSolve[eqn,y,x] **na k**ά**noume e**ύ**kola thn dokim**ή:

```
lysh1 = soll0@@1, 1DD
eqn ê. lysh1
```

```
y → Function@8x<, Ą^(-1) ê x C@1DD
```

```
True
```

8.2 Oi dunatóthteV thV DSolve

Genikά gia thn DSolve **prépei na xéroume óti epilúei** 1) ó**leV tiV gram**. **exisώseiV me staqeroúV suntelestéV poiasdήpote tάxhV** 2)é**na eurú jάsma grammikών exisώsewn me mh staqeroúV suntelestéV mécri** 2**hV tάxhV kai** 3) é**na eurú jάsma mh grammikών diajorikών exisώsewn**. **Mporeí epíshV na epilúsei diajorikéV exisώseiV me merikéV paragώgouV arkei na dώsoume tiV anexάrthteV metablhtéV apo tiV opoíeV exartάtai h zhtoúmenh sunάrthsh**, **upó morjή lístaV**. **Akolouqoún paradeígmata**:

Grammikή diajorikή exíswsh 1**hV tάxhV**

```
eqn1 = y '@xD + y@xD ê x + Cos@xD - E^x / x « 0
soll1 = DSolve@eqn1, y, xD
```

$$-A^x + \frac{\text{Cos}[x]}{x} + \frac{y[x]}{x} + y'[x] == 0$$

```
99y → FunctionA8x<, C@1D / x + (A^x - Sin@xD) / xE==
```

Eύresh miaV timήV thV y **p.c thV** y[3]

```
y@3D ê. soll1@@1, 1DD
```

$$\frac{C[1]}{3} + \frac{1}{3}\left(A^3 - \text{Sin}[3]\right)$$

Ό**tan upάrcoun arcikéV sunqήkeV tiV anagrάjoume mésa sthn entolή** DSolve **me morjή lístaV sth morjή**:

```
DSolve[{eqn, sunqήkeV},y,x]
```

P. **c gia na luqeí h** eqn1 **me thn arcikή sunqήkh** y[2]==1 **tóte qa grάyoume**:

```
DSolve@8eqn1, y@2D « 1<, y, xD
```

$$\{\{y \to \text{Function}\left[\{x\}, \frac{2 - A^2 + A^x + \text{Sin}[2] - \text{Sin}[x]}{x}\right]\}\}$$

Grammikή diajorikή exíswsh 2hV táxhV me staqeroúV suntelestéV

```
eqn2 = y''@xD - 2 y'@xD + y@xD « 0
soll1 = DSolve@eqn2, y, xD
```

$$y[x] - 2\, y'[x] + y''[x] == 0$$

$$\{\{y \to \text{Function}[\{x\}, e^x C[1] + e^x x C[2]]\}\}$$

Edώ emjanízontai dúo staqeréV sthn lúsh. Gia na exajanísoume thn mia apo tiV dúo qa prépei na dώsoume kápoieV arcikéV sunqήkeV sthn y ή **sthn parágwgo** y'. **An dώsoume arcikéV sunqήkeV sugcrónwV kai stiV dúo tóte paírnoume mia lúsh** y **cwrίV staqeréV**:

```
DSolve@8eqn2, y@0D « 1<, y, xD
DSolve@8eqn2, y'@1D « 0<, y, xD
DSolve@8eqn2, y@0D « 1, y'@1D « 0<, y, xD
```

$$\{\{y \to \text{Function}[\{x\}, e^x (1 + x) C[2]]\}\}$$

$$\{\{y \to \text{Function}[\{x\}, e^x (-2 + x) C[2]]\}\}$$

$$\{\{y \to \text{Function}\left[\{x\}, -\frac{1}{2} e^x (-2 + x)\right]\}\}$$

Grammikή diajorikή exíswsh 2hV táxhV me mh staqeroúV suntelestéV

```
eqn3 = x y''@xD - H2 x + 1L y'@xD + Hx + 1L y@xD « x è!!!x E^x
DSolve@eqn3, y, xD
```

$$(1 + x)\, y[x] - (1 + 2x)\, y'[x] + x\, y''[x] == e^x x^{3/2}$$

$$\left\{\left\{y \to \text{Function}\left[\{x\}, \frac{4}{5} e^x x^{5/2} + e^x C[1] + \frac{1}{2} e^x x^2 C[2]\right]\right\}\right\}$$

Exíswsh me merikéV paragώgouV wV proV x **kai** y **kai ágnwsth sunárthsh thn** z[x,y]

```
eqn4 = x D@z@x, yD, xD + y D@z@x, yD, yD « z@x, yD
soll4 = DSolve@eqn4, z, 8x, y<D

y z^H0,1L@x, yD + x z^H1,0L@x, yD == z@x, yD
```

```
99z → FunctionA8x, y<, x C@1DA y/x EE==
```

Prosocή edώ me C@1D@ y/x D**ennooúme mia sunárthsh** C[1] **miaV metablhtήV éstw** t **sthn opoía écoume antikatastήsei thn metablhtή** t **me to lógo** y/x. **AV kánoume kai dokimή**

```
soll4@@1, 1DD
eqn4 ê. soll4@@1, 1DD
eqn4 ê. soll4@@1, 1DD êê Simplify

z → FunctionA8x, y<, x C@1DA y/x EE
```

```
y C@1D'A y/x E + x (C@1DA y/x E − y C@1D'@ y/x D / x) == x C@1DA y/x E
```

```
True
```

H Simplify **kánei touV apaitoúmenouV upologismoúV kai aplopoieí thn parástash**

OmogenήV grammikή exíswsh 2**hV táxhV** (**me staqeroúV suntelestéV kai**)**me merikéV paragώgouV kai ágnwsth sunárthsh thn** z[x,y]

```
eqn5 =
 D@z@x, yD, 8x, 2<D − 5 D@z@x, yD, x, yD + 6 D@z@x, yD, 8y, 2<D « 0
soll5 = DSolve@eqn5, z, 8x, y<D

6 z^H0,2L@x, yD − 5 z^H1,1L@x, yD + z^H2,0L@x, yD == 0
```

```
88z → Function@8x, y<, C@1D@3 x + yD + C@2D@2 x + yDD<<
```

Edώ blépoume óti h genikή lúsh periécei duo sunartήseiV C[1] **kai** C[2] **stiV opoíeV écoume antikatastήsei thn metablhtή touV me** 3 x+y **kai me** 2 x+y **antístoica**. H **dokimή gínetai eúkola**:

```
soll5@@1, 1DD
eqn5 ê. soll5@@1, 1DD
eqn5 ê. soll5@@1, 1DD êê Simplify

z → Function@8x, y<, C@1D@3 x + yD + C@2D@2 x + yDD
```

$$9\,C@1D^{\grave{A}\grave{A}}@3\,x+yD+4\,C@2D^{\grave{A}\grave{A}}@2\,x+yD+6\,HC@1D^{\grave{A}\grave{A}}@3\,x+yD+C@2D^{\grave{A}\grave{A}}@2\,x+yDL-5\,H3\,C@1D^{\grave{A}\grave{A}}@3\,x+yD+2\,C@2D^{\grave{A}\grave{A}}@2\,x+yDL==0$$

```
True
```

8.3 Sustήmata diajorikών exisώsewn kai h entolή FullSimplify

Me thnDSolve **mporoύme na epilύsoume kai sustήmata diajorikών exisώsewn, aplά qa prosέxoume na dώsoume mazί me tiV diajorikέV exisώseiV kai tiV arcikέV ή oriakέV sunqήkeV mέsa se mia lίsta kai tiV ágnwsteV sunartήseiV mέsa se állh lίsta.P.c**

```
system6 =
  8y'@xD + y@xD - z@xD « Sin@xD, z'@xD + y@xD + z@xD « Cos@xD<;
agnwstesSynarthseis6 = 8y@xD, z@xD<;
soll6 = DSolve@system6, agnwstesSynarthseis6, xD
soll6 êê Simplify
soll6 êê FullSimplify

88y@xD → Ą^-x C@1D Cos@xD + Ą^-x C@2D Sin@xD +
    Ą^-x Sin@xD HĄ^x Cos@xD^2 + Ą^x Sin@xD^2L, z@xD → Ą^-x C@2D Cos@xD -
    Ą^-x C@1D Sin@xD + Ą^-x Cos@xD HĄ^x Cos@xD^2 + Ą^x Sin@xD^2L<<
```

$$88y@xD\to Ą^{-x}\,HC@1D\,Cos@xD+HĄ^{x}+C@2DL\,Sin@xDL,\;z@xD\to Ą^{-x}\,HHĄ^{x}+C@2DL\,Cos@xD-C@1D\,Sin@xDL<<$$

$$88y@xD\to Sin@xD+Ą^{-x}\,HC@1D\,Cos@xD+C@2D\,Sin@xDL,\;z@xD\to Cos@xD+Ą^{-x}\,HC@2D\,Cos@xD-C@1D\,Sin@xDL<<$$

Edώ crhsimopoihsame kai thn sunárthsh FullSimplify **gia na kánoume óso to dunatón perissótereV aplopoihseiV. AlliώV h lush ópwV blépoume écei perίplokh morjή. To ídio qa kánoume ópou creiázetai kai parakátw. Gia thn dokimή qa prépei na broύme kai tiV paragώgouV** y'[x] **kai** z'[x] **kai na antikatastήsoume**:

```
system6 ê. soll6@@1DD ê. D@soll6@@1DD, xD
system6 ê. soll6@@1DD ê. D@soll6@@1DD, xD êê Simplify
```

```
8Ą^-x Sin@xD HĄ^x Cos@xD^2 + Ą^x Sin@xD^2L == Sin@xD,
 Ą^-x Cos@xD HĄ^x Cos@xD^2 + Ą^x Sin@xD^2L == Cos@xD<
```

```
8True, True<
```

Scólio An eícame qései parapánw agnwstesSynarthseis6={y, z} **antí** agnwstesSynarthseis6={y[x],z[x]} **qa eícame apojúgei na broúme kai tiV paragώgouV twn** y **kai** z **gia na kánoume thn dokimή** . **Autó ómwV den to káname dióti h apánthsh pou qa maV dinótan qa perieíce wV gnwstó, ta** yØFunction[{x},... **kai** zØFunction[{x},...**kai dustucώV tóte to** *Mathematica* **adunateí na kánei** Simplify ή FullSimplify!!!!

An sto parapánw sύsthma écoume kai arcikéV sunqήkeV p.c y[**p**/2]=1 **kai** z[**p**/2]=0 **tóte autéV prostíqontai polú aplά sto sύsthma**:

```
DSolve@8system6, y@Pi ê 2D « 1, z@Pi ê 2D « 0<,
  agnwstesSynarthseis6, xD êê FullSimplify
```

```
88y@xD → Sin@xD, z@xD → Cos@xD<<
```

8.4 H entolή NDSolve

Kleínoume to kejάlaio me thn NDSolve **h opoía parécei th dunatόthta ariqmhtikήV epílushV diajorikών exisώsewn kai susthmάtwn. SunήqwV crhsimopoioúme thn** NDSolve ó**pou mia akribή lύsh den mporeí na breqeí me thn** DSolve. **P.c**

```
Clear@eqn, sollD
eqn = y''@xD + 5 Log@y@xDD « 0;
soll « DSolve@8eqn, y'@0D « 1, y@0D « 1<, y@xD, xD
```

```
DSolve::bvimp :
 General solution contains implicit solutions. In the boundary value problem
   these solutions will be ignored, so some of the solutions will be lost.
```

```
soll == 8<
```

DSolve::bvimp: General solution contains implicit solutions. In the boundary value problem these solutions will be ignored, so some of the solutions will be lost.

soll=={}

H **sύntaxh thV entolήV eínai**

NDSolve[**diajorikéV exisώseiV**,y[x],{x,xmin,xmax}]

An upárcoun perissótereV apo mia diaj. **exisώseiV mpaínoun se lísta mazí me tiV arcikéV sunqήkeV an autéV bébaia upárcoun**. **Anti** y[x] **mporoúme na gráyoume pio aplá** y. **Fusiká ópwV ήdh xéroume sthn deúterh períptwsh qa pároume thn** y **wV sunárthsh tou** x **enώ sthn prώth períptwsh qa pároume thn timή** y[x] **thV** y **se morjή kanóna**. **Sto téloV prépei na dhlώsoume kai to diásthma** {x,xmin,xmax}, **mésa sto opoío qéloume na doqeí h** (**proseggistikή**) **lύsh**. **P.c**

```
Clear@sollD
soll = NDSolve@8eqn, y'@0D « 1, y@0D « 1<, y, 8x, 0, 4<D
soll1 = NDSolve@8eqn, y'@0D « 1, y@0D « 1<, y@xD, 8x, 0, 4<D

88y → InterpolatingFunction@880., 4.<<, <>D<<
```

```
88y@xD → InterpolatingFunction@880., 4.<<, <>D@xD<<
```

H NDSolve **energeí wV exήV**: **Upologízei th lύsh** y, **proseggistiká**, **prώta gia éna mikró plήqoV shmeíwn tou diastήmatoV** {xmin,xmax}. **Autá ta shmeía mazí me tiV prokúptouseV timέV ta bázei se mia lísta pou thn onomázei gia suntomía** <>. **Sth sunéceia kánei katállhlh parembolή stiV parapánw timέV thV lístaV** <> **gia na mporései na breí thn timή thV** y[x] **gia opoiodήpote állo shmeío** x **tou diastήmatoV**. **Gia autó állwste sthn apánthsh h sunárthsh** y **anajéretai wV** Interpolating-Function **dhladή mia sunárthsh pou oi timέV thV** y[x] **upologízontai me thn méqodo parembolήV**. **An qéloume na broúme gia parádeigma thn timή** y[0.02] **qa gráyoume éna apo ta parakátw**

```
Hy ê. soll@@1, 1DDL@0.02D
soll@@1, 1, 2DD@0.02D
y@xD ê. soll1 ê. x → 0.02

1.01999
```

```
1.01999
```

```
81.01999<
```

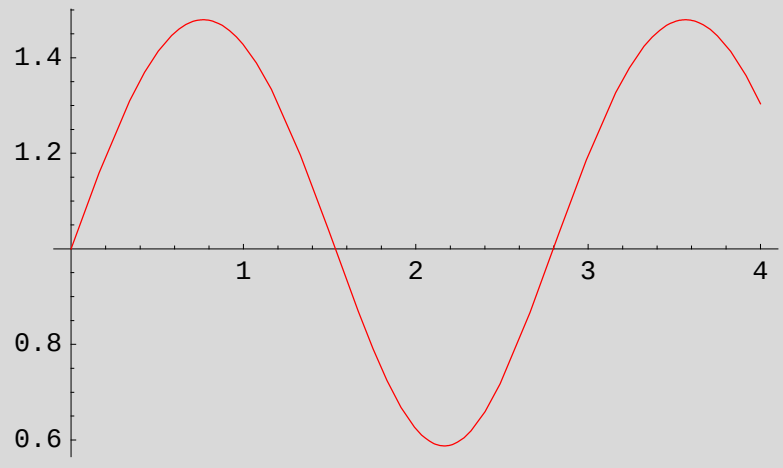
Prosocή den gráyame skéta y/.soll[[1,1]][0.02] **ή** y/.soll[[1]]/.x->0.02 **dióti den qa doulέyei**! **Genikά me** InterpolatingFunction[…][*x*] **brískoume thn timή thV sunárthshV parembolήV** y **sto** x. **EpίshV mporoύme na broúme kai óseV álleV timέV qáloume kai na kánoume kai grajikή parástash thV** (**proseggistikήV**) **lύshV thV diajorikήV exίswshV**. **P.c an qeloume na broúme tiV timés** y[x] **gia** x=0, 0.08, 2 0.08, 3 0.08, ... **qa gráyoume**

```
pinakas1 = Table@y@xD ê. soll1, 8x, 0, 4, .08<D
pinakas = Table@soll@@1, 1, 2DD@xD, 8x, 0, 4, .08<D

881.<, 81.07958<, 81.15673<, 81.22925<, 81.29518<, 81.35286<,
 81.40089<, 81.43816<, 81.46382<, 81.47731<, 81.47834<,
 81.46688<, 81.44318<, 81.40776<, 81.36143<, 81.30524<,
 81.24056<, 81.169<, 81.09246<, 81.01311<, 80.933368<,
 80.855841<, 80.783302<, 80.718571<, 80.664395<, 80.623265<,
 80.597208<, 80.587575<, 80.594882<, 80.618737<, 80.657895<,
 80.710411<, 80.773845<, 80.845475<, 80.92248<, 81.00208<,
 81.08163<, 81.15869<, 81.23106<, 81.2968<, 81.35424<,
 81.40201<, 81.43898<, 81.46433<, 81.4775<, 81.4782<,
 81.46642<, 81.44241<, 81.4067<, 81.3601<, 81.30367<<
```

```
81., 1.07958, 1.15673, 1.22925, 1.29518, 1.35286, 1.40089,
 1.43816, 1.46382, 1.47731, 1.47834, 1.46688, 1.44318,
 1.40776, 1.36143, 1.30524, 1.24056, 1.169, 1.09246, 1.01311,
 0.933368, 0.855841, 0.783302, 0.718571, 0.664395, 0.623265,
 0.597208, 0.587575, 0.594882, 0.618737, 0.657895, 0.710411,
 0.773845, 0.845475, 0.92248, 1.00208, 1.08163, 1.15869,
 1.23106, 1.2968, 1.35424, 1.40201, 1.43898, 1.46433,
 1.4775, 1.4782, 1.46642, 1.44241, 1.4067, 1.3601, 1.30367<
```

Oi timév den diajéroun se káqe perípwsh aplóV ston prώto pínaka emjanízontai me thn morjή lístaV. Me thn Plot **mporoúme na kánoume thn grajikh thV parástash**:

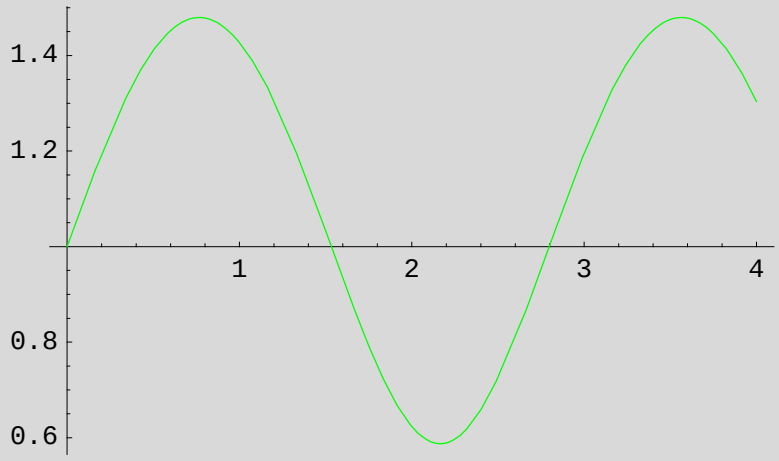
```
Plot@y@xD ê. soll1, 8x, 0, 4<, PlotStyle → RGBColor@1, 0, 0DD
```




```
¦ Graphics ¦
```

Fusiká gia thn deúterh perípwsh qa mporoúsame na gráyoume

```
Plot@soll@@1, 1, 2DD@xD, 8x, 0, 4<, PlotStyle → RGBColor@0, 1, 0DD
```




```
¦ Graphics ¦
```

An tώra kάnoume klík pάnw sto scήma kai sthn sunéceia patήsoume sunecώV to plήktro Alt **mporoύme na doύme sthn aristerή gwnía thV oqónhV maV tiV suntetagméneV twn shmeíwn thV kampýlhV maV**. **H entolή** NDSolve **ejarmózetai exísou apotelesmatikά kai se sustήmata diajorikών exisώsewn**.

Άskhsh: Na luqeí ariqmhtikά to sύsthma twn diajorikών exisώsewn y'[x]==Log[z[x]]+y[x],z'[x]==-Log[y[x]]-2 z[x]**me arcikέV sunqήkeV** y[2]==z[2]==1 **sto diάsthma** [2,5]. **Na breqoύn oi timέV twn sunartήsewn** y[x] **kai** z[x] **sta shmeía** x=2,2.5,3,3.5,...,5 **kai na gínoun oi grajikέV touV parastάseiV**.

Κεφάλαιο 9ο: Γραφικές Παραστάσεις

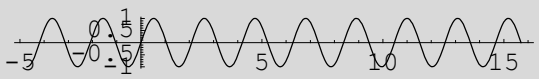
9.1 Διδιάστατες Γραφικές Παραστάσεις

9.1.1 GrajikέV parastάseiV kampulών.

Όtan qέloume na kάnoume thn grajikή parάstash miaV kampύlhV c prépei kat' arcήn na doύme me poió trόpo perigrάjetai h c. PeriptώseiV:

1. H kampύlh dínetai wV grάjhma miaV sunάrthshV f tou x. P.c f[x]=Cos[x]. Tóte mporoύme na crhsimopoiήsoume thn Plot:

```
Plot@Sin@3 xD, 8x, -3 Pi ê 2, 5 Pi<, AspectRatio → AutomaticD;
```



Scólia: PatώntaV pάnw sto grajikό me to pontíki o deíkthV se rómbo pou écei ena staurό mésa kai emjanίzetai éna plaίsio me labέV. Me to aristerό plήktro tou pontikioύ mporoύme na metakinήsoume to grajikό. Me tiV "labέV" mporoύme na alláxoume tiV diastάseiV tou grajikoύ, surontάV thV se diάjoreV kateuqύnseiV.

An qέloume na "zoumάroume" se kάpoio kommάti tou grajήmatoV qa mporoύsame na sύroume katάllhla tiV labέV patώntaV tautόcrona to Ctrl.

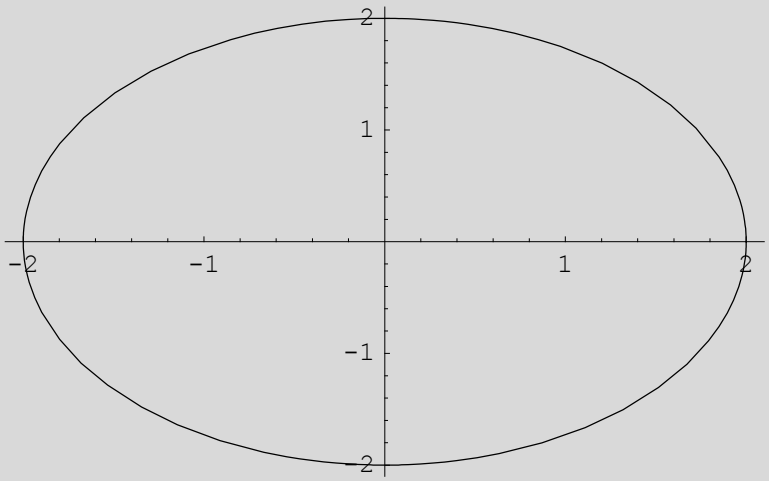
PatώntaV pάnw sto grάjhma kai sth sunέceia to Ctrl(sunecόmena)kai pάli klίk sto grάjhma mporoύme na doύme na schmatίzetai éna staurόV kai tiV suntetagméneV opoiodήpote shmeίou (ekeίnou pou shmadeύei to kέntro tou staurού) kάtw aristerά sto paráquro.

H Plot paírnei kάpoia shmeία tou x sto doqέn diάsthma (p.c {x,-3 Pi/2,5 Pi})kai crhsimopoiώntaV ta, upologίzei thn sunάrthsh y (edώ Sin[3 x]) kai me ta zeύgh (x,y) kάnei thn grajikή parάstash. Ta shméia x pou epilégontai légontai PlotPoints kai eínai líga. Gia autό upάrcoun kai oi atέleiV ston scediasmό thV kampύlhV. ProsέxteV tiV atέleieV gia parάdeigma ston scediasmό thV Sin[3 x] ótan megenqύnete trabώntaV apo tiV labέV. ΌmwV me PlotPoints->40 gia parάdeigma, mporeίtai na écete éna kalύtero apotέlesma.

Άskhsh: Máqete perissόtera gia tiV epilogέV Compiled, MaxBend, PlotPoints kai PlotDivision kai prospaqeίste na tiV ejarmόsete sto scediasmό thV Sin[3 x] me {x,0,4 Pi}. Gia na deíte kalύtera to problhma sto scediasmό qέste sthn Plot to AspectRatio->Automatic kai koitάxte thn kampύlh p.c sta baqoulώmata . Den eínai tόso omalή óso se álla shmeία thV. Gia thn AspectRatio qa milήsoume amέswV parakάtw.

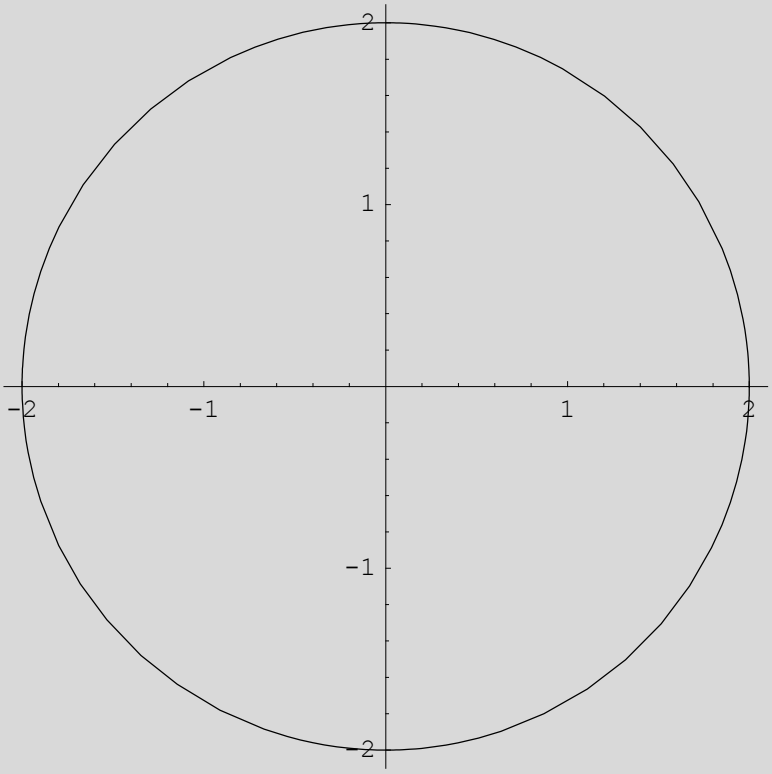
2. Όtan h kampύlh den eínai grάjhma sunάrthshV(p.c ótan h kampύlh eínai énaV kύkloV) tóte den mporoύme na crhsimopoiήsoume thn Plot. AV upoqέsoume loipón óti h kampύlh orίzetai me parametrikέV exisώseiV. P. c oi parametrikέV exisώseiV enόV kύklou me aktίna ίsh me r eínai x=r Cos[t], y=r Sin[t], me parάmetro t sto diάsthma [0,2 p]. Se tétoieV periptώseiV crhsimopoioύme thn entolή ParametricPlot

```
ParametricPlot@82 Cos@tD, 2 Sin@tD<, 8t, 0, 2 Pi<D;
```



Scólio: **To scήma pou proékuye moiázei me élleiyh kai óci me kúklo**! **Autó wjeíletai sto gegonóV óti autómata kaqorízetai** (**apo thn** Plot) **o lógoV tou úyouV tou plaisíou**(**pou peribállei to grájhma**) **proV to plátoV tou plaisíou na eínai** 1 **allá** 1/ **Crusή Tomή dhl**. **perípou** 61.8034 % **eínai mikrótero to úyoV apo to plátoV**! **An qéloume na emjanísoume to scήma ópwV pragmatiká eínai den écoume para na qésoume** AspectRatio->1(**uyoV dia plátoV**=1 **dhl**. **tetrágwno plaísio**) ή **kalútera** AspectRatio->Automatic (**dhl**. **h monáda métrhshV mήkouV ston** Ox =**monáda métrhshV ston** Oy):

```
ParametricPlot@82 Cos@tD, 2 Sin@tD<,
  8t, 0, 2 Pi<, AspectRatio → AutomaticD;
```

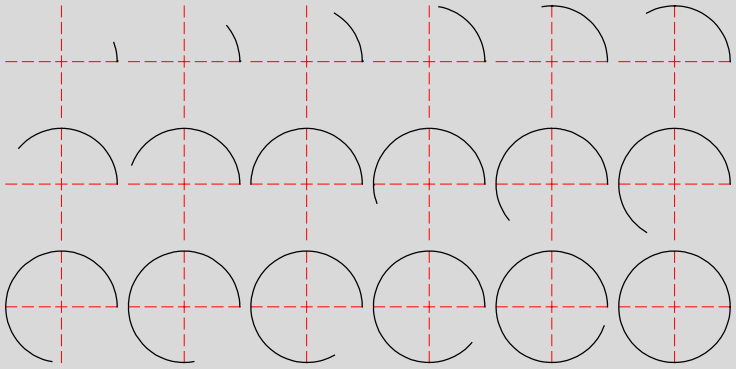


To AspectRatio**Ø**Automatic **eínai stiV perissótereV periptώseiV arketá kaló dióti den paramorjώnei thn kampúlh. Se me rikéV ómwV peritώseiV ótan gia parádeigma ótan écoume megálei dusanalogía tou mήkouV tou pedíou tou** x(**pedío orismoú**) **proV to mήkoV tou pedíou tou** y(**pedío timών**) **crhsimopoioúme mia sugkekrimménh timή tou** AspectRatio **ώste na écoume éna katanohtó grájhma**.

Άskhsh. **Scediάste thn kampúlh** 1/x **gia** {x,-,1,1} **kai** AspectRatio-> Automatic. T**i parathreíte**;. **Allάxte to** AspectRatio **p.c se** 1/2 **ή káti állo ή ajairéste to** AspectRatio **gia na beltiώsete to grájhma**.

Scólio: **KaqώV to** t **megalώnei apo** 0 **proV to** 2 **p ólo kai megalύtero méroV tou kύklou scediάzetai**. **Gia na doúme tiV diadocikéV "jáseiV" mporoúme na crhsimopoiήsoume thn entolή** Show **se sunduasmó me thn entolή** GraphicsArray **p.c**

```
akoloythia = Table@ParametricPlot@82 Cos@tD, 2 Sin@tD<, 8t, 0, n Pi<,
    DisplayFunction → Identity, Ticks → None, AspectRatio → 1,
    AxesOrigin → 80, 0<, PlotRange → 88-2, 2<, 8-2, 2<<, AxesStyle →
     8RGBColor@1, 0, 0D, Dashing@8.1, .05<D<D, 8n, 1 ê 9, 2, 1 ê 9<D;
kommatia = Partition@akoloythia, 6D;
Show@GraphicsArray@kommatiaDD;
```



H epilogή DisplayFunction**Ø**Identity **anagkάzei thn** ParametricPlot **na mhn scediάsei**. **H** akoloythia **periécei** 18 **grajikéV parastάseiV sth seirά kai h** Partition **kόbei thn** akoloythia **se kommάtia twn** 6 **grajhmatwn**. **Opóte sta** kommatia **qa écoume** 3 **grammέV epi** 6 **grajήmata to kaqéna**. **H** GraphicsArray **topoqeteί ta grajήmata tou pínaka sth seirά(karé - karé) se ómoia plaísia kai h** Show **emjanízei telikά to apotέlesma**! **Gia tiV epilogέV pou upάrcoun mέsa sto** ParametricPlot ó**pwV gia parádeigma** PlotRange**Ø**{{-2,2},{-2,2}} **écoun basikó skopó na kάnoun pio katanohtά kai elkustikά ta grajikά**. **Me thn** PlotRange **léme poiά shmeία epiqumoúme na emjanízontai sthn eikóna**. **Me** PlotRange-> All **emjanízontai ólα ta shmeία thV kampúlhV**.

Άskhsh: **Máqete perissótera gia tiV parapάnw epilogέV thV** ParametricPlot **mesw tou** Help **kai prospaqήsete na peiramatisteíte me autέV**.
Άskhsh: **Scediάsete thn** 3^{-x^2} **gia** {x,-5,5} **dialégontaV diajoretikά** PlotRange **kai** AspectRatio. **P.c qéste** PlorRange->All, PlotRange->{{-1,1},All}, PlotRange->{All,{-1,.02}},PlotRange->{-4,1}, PlotRange->{{2,10},{0,.02}} **k.o.k. kai diάjoreV epilogέV gia** AspectRatio **p.c** Automatic, 1/2 **k.o.k**

3. **H kampúlh dínetai wV grájhma miaV sunάrthshV, pou orízetai peplegména dhl. mέsw exíswshV**.
P.c gia na kάnoume thn grajikή parάstash thV parabolήV x^2+3 y==5 **sto diάsthma** [-3,3] **qa prépei na lύsoume wV proV** y (y=Sqrt[5- x^2]/3)**kai metά na crhsimopoiήsoume thn** Plot. **Kalύtera ómwV gia qa ήtan na kalέsoume thn entolή** ImplicitPlot **cwríV na creiasteί na kάnoume parapάnw douleiά**:

```
ImplicitPlot@x^2 + 3 y « 5, 8x, −3, 3<D

ImplicitPlot@x^2 + 3 y == 5, 8x, −3, 3<D
```

Dustucώς den maς έbgale tίpota! Ας doύme to lόgo. An patήsoume F1 **kai dώsoume thn lέxh** ImplicitPlot **qa diapistώsoume όti autή brίsketai sto pakέto** Graphics`ImplicitPlot`. **To** Graphics **eίnai h dieύqunsh** (folder)**mέsa sto opoίo brίsketai to pakέto** ImplicitPlot. **Mέsa sto pakέto autό έcei orisqeί h sunάrthsh** ImplicitPlot. **Pakέto lέme έna arceίo me touς orismoύς kάpoiwn sunartήsewn pou έcoun skopό na kάnoun mia eidikή ergasίa p.c na scediάsoun mia sunάrthsh όtan autή dίnetai se polikές suntetagmέneς k.o.k. Apo thn stigmή pou kaloύme έna pakέto den creiάzetai na to xanakalέsoume parakάtw gia deύterh jorά! Έna pakέto to kaloύme me thn entolή** Needs **P.c** Needs["Graphics`ImplicitPlot`"]. **Pio aplά sunήqwς grάjoume to diplό <<** sthn qέsh tou Needs **p.c**.

```
<< Graphics`ImplicitPlot`
```

```
ImplicitPlot::shdw : Symbol ImplicitPlot appears in multiple
   contexts 8Graphics`ImplicitPlot`, Global`<; definitions in context
   Graphics`ImplicitPlot` may shadow or be shadowed by other definitions.
```

Prosέxte amέswς maς έbgale έna mήnuma pou léei όti sunάnthse thn lέxh ImplicitPlot **se dύo diajoretikά** context **sto** Global **kai sto** Graphics`ImplicitPlot`. **To context eίnai to όnoma tou pakέtou. To** Global **eίnai έna genikό pakέto kai anoίgei autόmata kάqe jorά pou anoίgoume to** *Mathematica*. **Eίnai "o cώroς upodocής" se όpoion mpaίnei sto** *Mathematica*. **Gia parάdeigma kάqe jorά pou grάjome mia lέxh(p.c** ImplicitPlot) ή **έna sύmbolo miaς metablhtής, apoqhkeύetai sto** Global **ektός kai an emeiς tou zhtήsoume na to apoqhkeύsei se άllo arceίo. DustucώV h lέxh** ImplicitPlot **den ecei orisqeί sto** Global **me kάpoio trόpo. Den upάrcei oύte sto paketo tou sustήmatoς dhl. to System (sto** System **brίskontai oi orismoί twn pio basikών sunartήsewn όpwς p.c thς prόsqeshς, thς** Plot **k.o.k**). Έ**tsi to** *Mathematica* **qewreί thn** ImplicitPlot **san kάti néo kai den έcei idέa peri tίnoς prόkeitai** ή **poia sunάrthsh paristάnei!** Ό**tan kalέsoume to pakέto** <<Graphics`ImplicitPlot` **tόte to** *Mathematica* **anakalύptei mέsa s' autό xanά th sunάrthsh** ImplicitPlot **gi'autό paraponiétai!!! H** ImplicitPlot **paraménei mia sunάrthsh tou** Global **epeidή to** Global **episkiάzei to** (**eidikό**) **pakέto** Graphics`ImplicitPlot` Έ**tsi loipόn oύte tώra qa maς doulέyei h** ImplicitPlot **parόlo pou anoίxame to swstό pakέto**

```
ImplicitPlot@x^2 + 3 y « 5, 8x, −3, 3<D

ImplicitPlot@x^2 + 3 y == 5, 8x, −3, 3<D
```

Qa prέpei na diagrάyoume thn ImplicitPlot **apo to trέcon arceίo**(**edώ to** Global) **grάjontaς**:

```
Remove@ImplicitPlotD
```

Ας rwtήsoume tώra poieς sunartήseiς kai poia sύmbola brίskontai akόma mέsa sto Global

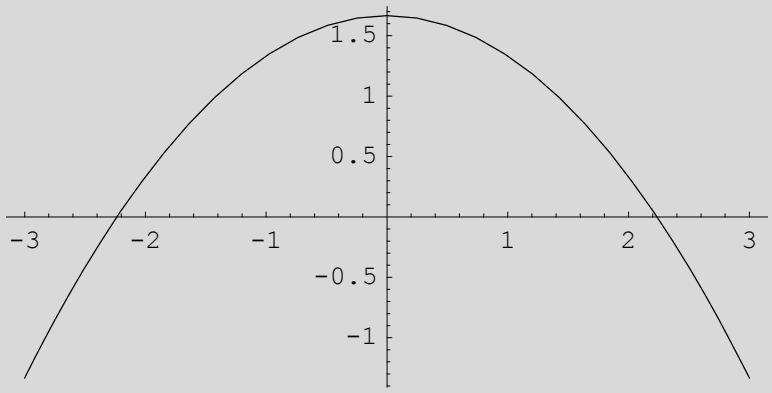
```
?Global`*
```

Global`

akoloythia kommatia n t x y

Όpwv blépoume den upárcei poiá h ImplicitPlot sto Global opóte aV thn kalésoume xaná

```
ImplicitPlot@x^2 + 3 y « 5, 8x, -3, 3<D;
```




```
¦ Graphics ¦
```

Blépoume loipón óti epeidή den bréqhke h ImplicitPlot sto Global, to *Mathematica* anagkásthke na yáxei (me kápoia sugkekrimménh seirá -autήn pou upárcei mésa sto $ContextPath) sta álla pakéta pou ήdh écoun anoicteí dhl. to System kai to Graphics`ImplicitPlot` kai jusiká brήke sto deύtero pakéto thn swstή sunárthsh kai thn ejármose!
Scólio: gia merikέV entolέV den creiázetai na kalésoume kápoio eidikó pakéto gia na ektelestoύn dióti brískontai sto System. Edώ mésa brískontai óleV oi gnwstέV sunartήseiV tou Kernel tou *Mathematica*. GrájontaV $Packages mporoύme na doύme poiá pakéta écoun jortwqeί mécri tώra. Me thn entolή Context[f] paírnoume to ónoma tou pakétou pou periécei thn sunárthsh f enώ me $Context brískoume to Context sto opoίo briskómaste autή thn stigmή. Me $ContextPath paírnoume thn seirά proteraiόthtoV pou qa prépei na yáxei to *Mathematica* gia na breί touV orismoύV twn sumbólwn kai twn sunartήsewn. Aristerótera eínai ekeίno pou anoίcqhke teleutaίo (apo emάV me to << ή ίswV kai apo to *Mathematica* cwríV na to antilhjqoύme) p.c

```
$Packages
```

```
8Utilities`FilterOptions`, Graphics`ImplicitPlot`, Global`, System`<
```

```
Context@PlotD
Context@PlusD
Context@ImplicitPlotD
$Context
$ContextPath

System`
```

```
System`
```

```
Graphics`ImplicitPlot`
```

```
Global`
```

```
8Graphics`ImplicitPlot`, Utilities`FilterOptions`, Global`, System`<
```

4. **H kampýlh dínetai wV grájhma miaV sunárthshV**, **pou orízetai se polikéV suntetagméneV**. **To pakéto** Graphics`Graphics` **prépei na anoicqeí káqe jorá pou creiazómaste thn** PolarPlot.

```
<< Graphics`Graphics`
PolarPlot@Cos@2 tD, 8t, 0, 2 Pi<, Background → RGBColor@0, 0, .5D,
  PlotStyle → 8RGBColor@1, 0, 0D, Thickness@.01D<D;
```

Me PlotStyle **kaqorízoume pwV qéloume na scediasteí h grajikή parástash enώ me** BackgroundØ-RGBColor[0,0,.5] **epiléxame éna baqý mplé gia jónto**. **To** R(=0) **sth léxh** RGBColor **eínai to kókkino to** G(=0) **to prásino kai to B**(=.5) **eínai to mplé**. **AV parathrήsoume óti autómata epilécqhke éna kitrinwpó crώma gia touV áxoneV gia lógouV antíqeshV**. **An den maV arései to crώma autó**, **tóte me** AxesStyle->..... **mporoýme na to allάxoume**.
An tώra qéloume na doýme tiV diadocikéV jáseiV scediasmoý thV kampýlhV se kínhsh qa grάyoume

```
pinakas = Table@PolarPlot@Cos@2 tD, 8t, 0, n Pi<D, 8n, 0.1, 2, 0.1<D;
```

An kánoume diplό klík pánw se mia apo tiV parapánw graj. **parastά seiV qa doýme tiV jáseiV scediasmoý**! **Epeidή ta scήmata den bgήkan ópwV akribώV periméname aV kánoume kápoieV allagéV stouV axoneV**

```
Clear@pinakasD
pinakas = Table@PolarPlot@Cos@2 tD, 8t, 0, n Pi<,
    PlotRange → 88-1, 1<, 8-1, 1<<, Ticks → NoneD, 8n, 0.1, 2, 0.1<D;
```

To ídio apotélesma me to parapánw qa pérname me thn Do **cwríV thn crήsh thV** Table. **Dokimάste to parakάtw**

```
Do@PolarPlot@Cos@2 tD, 8t, 0, n Pi<,
  PlotRange → 88-1, 1<, 8-1, 1<<D, 8n, 0.1, 2, 0.1<D
```

Án mikrάnoume to bήma 0.1 sto {n,0.1,2,0.1}qa écoume perissótera karé kai άra pio argή kínhsh. pc mporeíte na antikatastήsete {n,0.1,2,0.01} sto Do kai na dokimάsete xanά!

Άskhsh: H parametrikή exíswsh thV prohgoúmenhV kampúlhV eínai x[t]=Cos[2 t] Cos[t] kai y[t]=Cos[2 t] Sint[t]. Prospaqeíste na párete tiV diadocikéV jáseiV crhsimopoiώntaV thn ParametricPlot antí thV Polar-Plot kai tiV parametrikéV exisώseiV anti tiV polikéV.

5. H kampúlh dínetai wV éna peperasméno sύnolo shmeίwn data pou écoun prokύyei apo metrήseiV p.c qermokrasίaV ή píeshV k.o.k p.c

```
data = TableA9x, 1/x=, 8x, 1, 10<E
synarthsh = Interpolation@dataD
Plot@synarthsh@xD, 8x, 1, 10<,
  PlotRange → Automatic, AxesOrigin → 80, 0<D;
```

$$\{\{1, 1\}, \{2, \frac{1}{2}\}, \{3, \frac{1}{3}\}, \{4, \frac{1}{4}\}, \{5, \frac{1}{5}\}, \{6, \frac{1}{6}\}, \{7, \frac{1}{7}\}, \{8, \frac{1}{8}\}, \{9, \frac{1}{9}\}, \{10, \frac{1}{10}\}\}$$

```
InterpolatingFunction@881, 10<<, <>D
```

H sunάrthsh Interpolation kάnei parembolή sta shmeία twn data kai epistréjei mia omalή kampúlh pou pernaei apo ta shmeία autά.

Mporoύme na crhsimopoiήsoume kai thn ListPlot kai na enώsoume ta shmeία twn data me euqúgramma tmήmata allά to apotélesma den eínai tόso ikanopoihtikό:

```
ListPlot@data, PlotJoined → True,
  PlotRange → Automatic, AxesOrigin → 80, 0<D;
```

H epilogή PlotJoinedØTrue anagkάzei thn ListPlot na enώnei me euqúgramma tmήmata ta shmeία pou scediάzei! Eínai polύ crήsimh sthn perίptwsh pou ta data maV den paristάnoun shmeία miaV sunάrthshV. P.c me thn ListPlot mporoύme na zwgrajίsoume asteίeV jatsoύleV:

```
koryfes =
  880, 0<, 80.5, 1<, 80.7, 0.5<, 80.5, 0.3<, 80.6, 0.2<, 8.3, 0<<;
fatsoyla = ListPlot@koryfes, PlotJoined → True, PlotRange → Automatic,
   AxesOrigin → 80, 0<, DisplayFunction → IdentityD;
mati = Graphics@8PointSize@0.03D, RGBColor@1, 0, 0D,
    Point@80.54, 0.72<D<D;
Show@fatsoyla, mati, DisplayFunction → $DisplayFunctionD
```

Me thn Show mporoúme na doúme pollá grajήmata mazí. Qa milήsoume xana gia autήn parakátw. Me **DisplayFunctionÆIdentity apojúgame na scediάsoume to perígramma tou kejalioú enώ me** Display-Function**Ø**$DisplayFunction **epanajérame thn dunatóthta sto** Show **na maV emjanísei thn jatsoúla mazí me to kókkino máti. Me thn** Graphics **mporoúme na scediάsoume shmeía, dískouV,euqúgramma tmήmata, polúgwna,na prosqétoume keímeno kai pollά álla**. Éna koítagma **sto** Help **tou** Graphics **qa saV peísei**...
Άskhsh: **CrhsimopoiώntaV thn sunάrthsh** Polygon **gia tiV** koryfes **na zwgrajísete xanά thn jatsoúla allά me to eswterikó tou kejalioú na eínai mplé** (**kai to máti kókkino**).

6. **An h kampúlh dínetai wV h sunάrthsh pou ikanopoieí kάpoia diajorikή exίswsh qa prépei prώta na lύsoume thn diajorikή exίswsh. P.c**

```
eqn = y''@xD + 5 Log@y@xDD « 0
soll = NDSolve@8eqn, y'@0D « 1, y@0D « 1<, y@xD, 8x, 0, 4<D
Plot@y@xD ê. soll, 8x, 0, 4<, PlotStyle → RGBColor@1, 0, 0DD;

5 Log@y@xDD + y''@xD == 0
```

```
88y@xD → InterpolatingFunction@880., 4.<<, <>D@xD<<
```

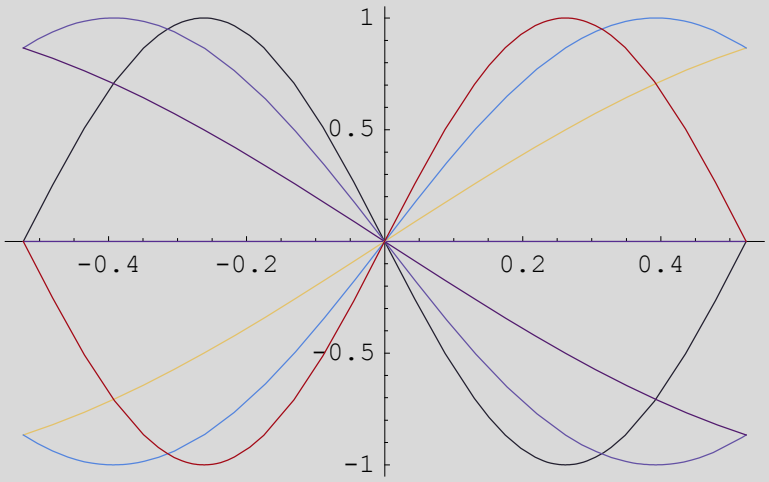
9.1.2 ScediάzontaV mia lίsta apo kampύleV

Me thn entolή Show **mporoúme na sunduάsoume grajikέV parastάseiV pou écoun gínei me diajoretikά pakéta**:

```
trigX@x_D := 88x, -0.03<, 8x, 0.03<, 8x + 0.03, 0<<
trigY@y_D := 88-0.03, y<, 80.03, y<, 80, y + 0.03<<
g1 = Graphics@8RGBColor@1, 0, 0D, Polygon@trigX@1.5DD,
    Polygon@trigY@1DD<, DisplayFunction → IdentityD;
g2 = Plot@Sin@3 xD, 8x, -Pi ê 6, Pi ê 6<, DisplayFunction → IdentityD;
g3 = PolarPlot@Cos@2 tD, 8t, 0, 2 Pi<, DisplayFunction → IdentityD;
Show@g1, g2, g3, Axes → True, PlotRange → 88-1, 1.53<, 8-1, 1.03<<,
  DisplayFunction → $DisplayFunctionD;
```

Me thn boήqeia thV Polygon **thV** Graphics **scediάzoume duo trigwnάkia éna sto shmeío** x **tou άxona O**x **kai éna sto shmeío** y **tou άxona O**y **antίstoica.Oi suntetagméneV touV eínai** trigX[x] **kai** trigY[y] **antίstoica**. **H** Show **sunduάzei óleV autέV mazí tiV grajikέV parastάseiV**.
An bέbaia eícame grajikέV parastάseiV pou scediάzontai me thn ídia entolή tóte jusikά den eínai anάgkh na crhsimopoiήsoume thn Show.
AV **kάnoume gia parάdeigma tiV grajikέV parastάseiV twn** Sin[n x] **gia** {n,-6,6,2} **dhl**. **gia** n=-6,-4,-2,0,2,4,6 **me thn** Show **kai cwrίV autήn**

```
grafika = Table@Plot@Sin@n xD,
    8x, -Pi ê 6, Pi ê 6<, DisplayFunction → Identity, PlotStyle →
     RGBColor@Random@ D, Random@ D, Random@DDD, 8n, -6, 6, 2<D;
Show@grafika, DisplayFunction → $DisplayFunctionD;
```



Dώsame éna tucaío crώma se káqe grajikή parástash gia na pároume éna apotélesma pio elkustikó. To Random[ ] **na upenqumήsoume epistéjei éna tucaío pragmatikó metaxú** 0 **kai** 1 **opóte me** RGBColor[Random[ ],-Random[ ],Random[]] **dώsame éna entelώV tucaío crώma sthn kampúlh maV** Sin[n x].

Állov trópoV cwríV thn Show:

Prώta qa mazeyoume ta hmítona se éna pínaka kai metá qa kánoume thn grajikή parástash tou pínaka autoú me thn Plot:

```
Remove@pinakasD
pinakas = Table@Sin@n xD, 8n, -6, 6, 2<D
Head@pinakasD
style =
  Table@8RGBColor@Random@ D, Random@ D, Random@DD<, 8n, -6, 6, 2<D;
Plot@pinakas, 8x, -Pi ê 6, Pi ê 6<, PlotStyle → styleD;

8-Sin@6 xD, -Sin@4 xD, -Sin@2 xD, 0, Sin@2 xD, Sin@4 xD, Sin@6 xD<
```

```
List
```

```
– Plot::plnr : pinakas is not a machine-size real number at x = -0.523599.

– Plot::plnr : pinakas is not a machine-size real number at x = -0.481117.

– Plot::plnr : pinakas is not a machine-size real number at x = -0.434787.

– General::stop :
   Further output of Plot::plnr will be suppressed during this calculation.
```

Blépoume autó den apédwse! O lógoV eínai óti h Plot **den mporeí na scediásei mia lísta apo sunartήseiV! Ára qa prépei me kápoio trópo na upologísoume prώta tiV sunartήseiV tou** pinakas **gia óleV tiV timέV tou** {n,-6,6,2}, **mia mia xecwristá, kai metá na analábei h** Plot **na zwgrajísei me thn seirá, káqe mia ap' autέV. Autó akribώV kánei h Evaluate. Allázei thn seirá twn entolών(twn energeiών). Parakátw kalésame kai to paketo** Graphics`Legend` **giatí qéloume na eiságoume kai éna pínaka me epexhghmatikέV etikέteV**:

```
<< Graphics`Legend`
etiketes = Table@ToString@Sin@n xDD, 8n, -6, 6, 2<D;
Plot@Evaluate@pinakasD, 8x, -Pi ê 6, Pi ê 6<, PlotStyle → style,
  PlotRange → 88-0.6, 0.6<, 8-1, 1<<, PlotLegend → etiketes,
  LegendShadow → 80, 0<, LegendSize → 8.6, .5<,
  LegendPosition → 80.8, -.6<, LegendBackground → GrayLevel@0.8DD;
```

Scólia:

a)Prosέxte mésa sto style **qa prépei na upárcoun** 7 **akribώV perigrajέV óseV kai oi sunartήseiV pou upárcoun sto** pinakas. **An upήrce móno mía p.c** PlotStyle**Ø**RGBColor[1,0,0] **tóte qa bgaínane óleV kókkineV!An upήrcan dύo perigrajέV p.c** PlotStyle**Ø**{RGBColor[1,0,0],RGBColor[0,1,0]} **tóte qa bgálei tiV "misέV" me prásino kai tiV álleV me kókkino!! Genikά mporeíte na jtiάxete to** style **ópwV eseíV qélete. P.c me** PlotStyle**Ø**{{RGBColor[1,0,0], Dashing[{.05,.02}]}, {GrayLevel[0.5]},{Hue[.03]},....} **qa pároume thn prώth kókkinh kai diakekkoménh thn deύterh sunárthsh mia apócrwsh tou gkrízou, thn tríth mia apócrwsh tou** Hue **k.o.k**

b)An den qélete ton pínaka diagráyte óti carakthristikó periécei thn léxh Legend. **An ómwV saV endiajérei ti akribώV kánei tóte peiramatisteíte me ta diájora carakthristiká thV** Legend **kai párte boήqeia apo to** Help! **H entolή** ToString[Sin[n x]] **epistréjei thn léxh** "Sin[n x]" . **ÓmwV epeidή p.c** Sin[-6 x] **eínai íso me** -Sin[6 x] **gia autó epistréjei autá pou blépete sth pinakída. Dokimáste** ToString[Sin[ToString[n x]] **an den saV arései to parapánw apotélesma.**

Áskhsh: Dínetai h élleiyh x^2+5 y^2==9, **kai h kampúlh pou écei parametrikή exíswsh** x[t_]:=Sin[t] **kai** y[t_]:=Sin[2 t] **gia** t **sto diásthma** {t,-4,4}. **Na scediásete thn élleiyh se kókkino crώma me thn** ImplicitPlot, **thn parametrikή exíswsh me thn** ParametricPlot **kai me diakekoménh grammή (p.c qéste** PlotStyle->Dushing[{.1,.05}]) **kai me thn** Show **zwgrajíste autέV apo koinoύ. Qéste epíshV sthn** Show Background->GrayLevel[.4]

Áskhsh: **Prospaqeíste na scediásete thn parágwgo** (D[x Cos[x],x]) **thV** x Cos[x] **gia** x **apo** 0 **mécri** 10. **Prosέxte na crhsimopoiήsete prώta thn** Evaluate! **AlliώV qa écete próblhma. H** Plot **den xérei pwV na zwgrajízei thn parágwgo miaV sunárthshV**!

9.1.3 Oi epilogέV twn didiástatwn grajikών.

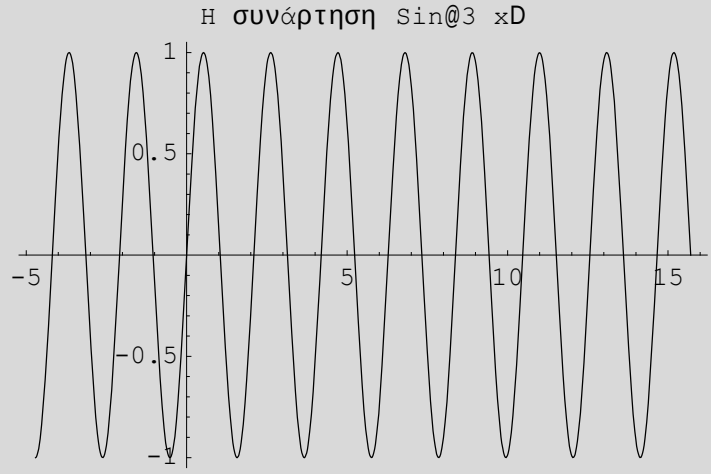
Me Options[ ] **mporoύme na doύme tiV epilogέV miaV opiasdήpote sunartήshV. P.c**

```
Options@PlotD

9AspectRatio → 1/GoldenRatio , Axes → Automatic,
 AxesLabel → None, AxesOrigin → Automatic, AxesStyle → Automatic,
 Background → Automatic, ColorOutput → Automatic, Compiled → True,
 DefaultColor → Automatic, Epilog → 8<, Frame → False,
 FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic,
 GridLines → None, ImageSize → Automatic, MaxBend → 10.,
 PlotDivision → 30., PlotLabel → None, PlotPoints → 25,
 PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
 Prolog → 8<, RotateLabel → True, Ticks → Automatic,
 DefaultFont Ç $DefaultFont, DisplayFunction Ç $DisplayFunction,
 FormatType Ç $FormatType, TextStyle Ç $TextStyle=
```

Edώ emjanízontai oi proepilegméneV timéV stie epilogéV. P.c h preoep. timή thV AspectRatio **eínai** AspectRatio → 1/GoldenRatio , **twn axónwn eínai A**xes**Ø**Automatic **k.o.k Creiazómaste éna biblío gia na analúsoume diexodiká káqe mia apo tiV parapánw epilogéV. Qa anajerqoúme me suntomía se kápoieV ap'autéV pou den écoume deí éwV tώra**.

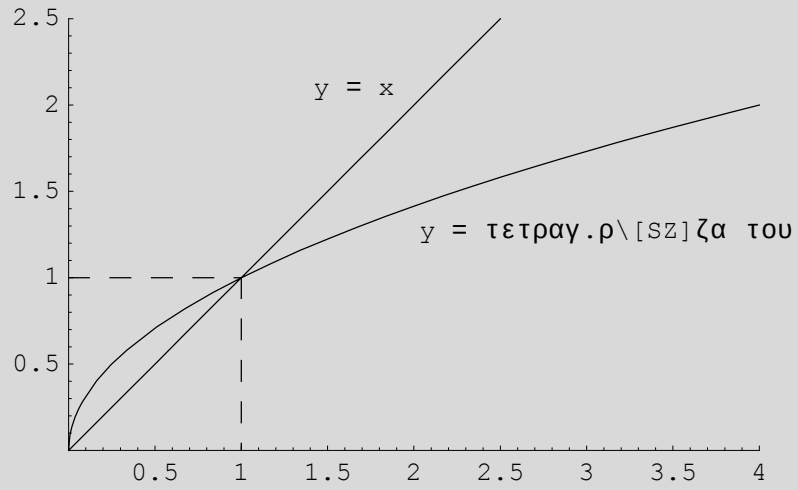
a) **Epigrajé**V(Labels): **Mporoúme na prosqésoume kápoia epigrajή me thn PlotLabel**

```
synhmitono = Plot@Sin@3 xD,
   8x, -3 Pi ê 2, 5 Pi<, PlotLabel -> "H συνάρτηση Sin@3 xD"D;
```



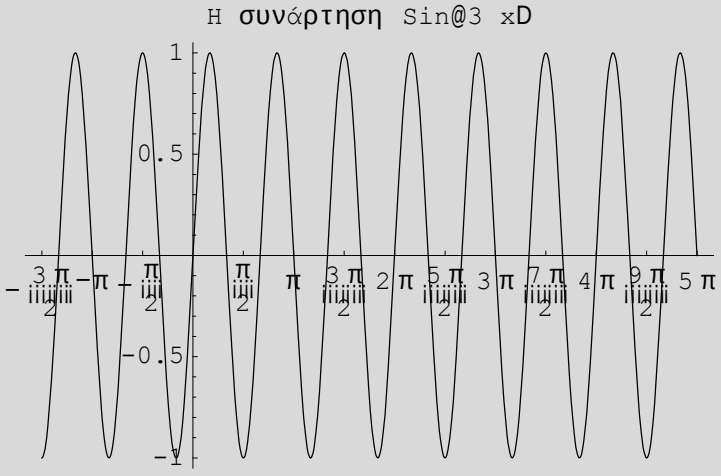
b) Epilog. **Mporoúme na crhsimopoiήsoume kai thn** FrameLabel ή **thn** AxesLabel(**lígo parakátw qa tiV doúme perissótero analutiká**) ή **thn Epilog gia ton ídio skopó. H** Epilog **eínai o** "**epílogoV**" **sto grájhmá maV dhl**. **kápoia stoiceía** (**p.c keímeno**,**etikéta**,**epiplέon shmeía**) **pou qéloume na mpoun ajótou oloklhrwqeí h grajikή parástash**. **Sto epómeno paráδeigma qésame ta keímena** "y=x","y=**tetr.ríza tou** x", **kaqώV kai tiV diakekommέneV proV touV axoneV apo to shmeío** (1,1)

```
Plot@8x, Sqrt@xD<, 8x, 0, 4<,
AspectRatio -> Automatic,
PlotRange -> 880, 4<, 80, 2.5<<,
Epilog -> 8 Text@"y = x", 82, 2<, 81.3, -1<D,
          Text@
      "y = τετραγ.ρ\\@SZDζα του x", 82, Sqrt@2D<, 8-1, 1.4<D,
          Dashing@80.03<D, Line@880, 1<, 81, 1<<D,
          Dashing@80.03<D, Line@881, 1<, 81, 0<<D <D;
```



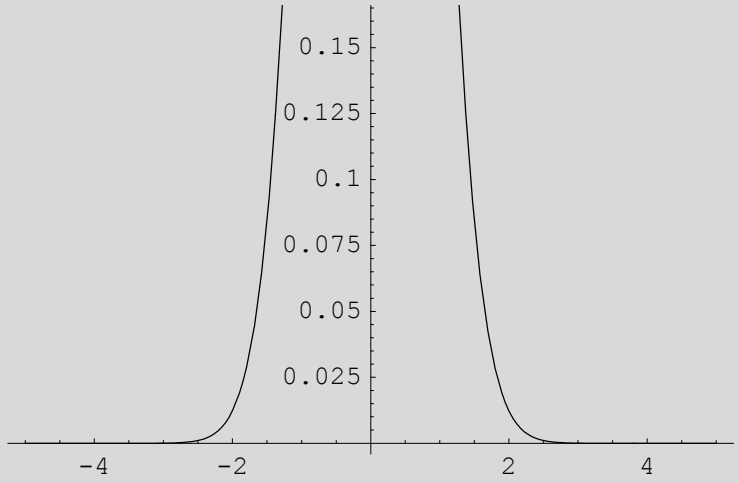
g)Ticks. M**poro**ύ**me na b**ά**loume Ticks sta sh**me**ía pou eme**í**V q**é**loume**. **Parak**ά**tw b**ά**zoume ston** ά**xona** Ox Ticks **sta sh**me**ía apo** -3 Pi/2 é**wV** 5 Pi **me b**ή**ma** Pi/2. To Ticks**Ø**{units,Automatic} **shma**í**nei** ó**ti ston** Oy **qa akolouqhqe**í **h proepilegm**é**nh akolouq**í**a**(**h** Automatic **tou** *Mathematica*) **twn** Ticks **en**ώ **ston** Ox **kata thn diki**ά **maV** units

```
units = Range@-3 Pi ê 2, 5 Pi, Pi ê 2D;
Show@synhmitono, Ticks → 8units, Automatic<D;
```



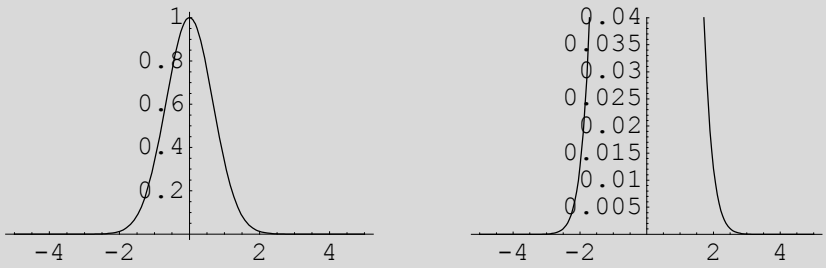
δ)**PlotRange**. Με PlotRenge->Automatic καθορίζεται αυτόματα απο το *Mathematica* ποιές τιμές y της καμπύλης θα εμφανιστούν στο τελικό γράφημα. Σε μερικές περιπτώσεις αυτό δεν πετυχαίνει με αποτέλεσμα να χάνουμε σημαντικές πληροφορίες. Π.χ

```
Plot@3^-x^2, 8x, -5, 5<D;
```



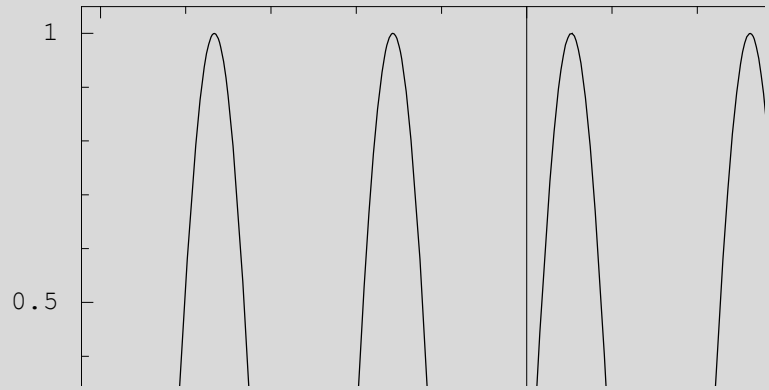
Το σχήμα έχει αποκοπεί στην γειτονιά του 0. Σε τέτοιες περιπτώσεις καλό είναι να βάζουμε **PlotRenge->All** έτσι ώστε να εμφανίζονται όλες οι τιμές της καμπύλης. Αν πάλι δεν έχουμε το επιθυμητό αποτέλεσμα ή θέλουμε να εμφανίζονται κάποιες συγκεκριμμένες τιμές του y, θα μπορούσαμε να βάλουμε ένα συγκεκριμμένο διάστημα **PlotRenge**->{ymin,ymax}π.χ

```
gr1 = Plot@3^-x^2, 8x, -5, 5<,
   PlotRange → All, DisplayFunction → IdentityD;
gr2 = Plot@3^-x^2, 8x, -5, 5<, DisplayFunction → Identity,
   PlotRange → 80, 0.04<D;
Show@GraphicsArray@8gr1, gr2<DD;
```



ε).**Me PlotRegion mporoúme na qésoume periqώria gúrw apo éna grάjhma**. **Me katάllhleV arnhtikéV timéV sto** PlotRegion **mporoúme na zoumάroume p.c**

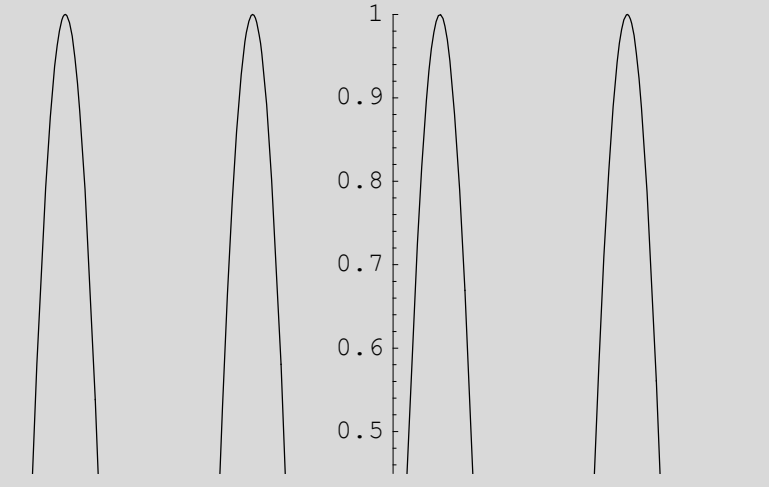
```
Plot@Sin@3 xD, 8x, -3 Pi ê 2, 5 Pi<,
 Frame → True, PlotRegion → 88-0.5, 3<, 8-1.7, 0.8<<D
```




```
¦ Graphics ¦
```

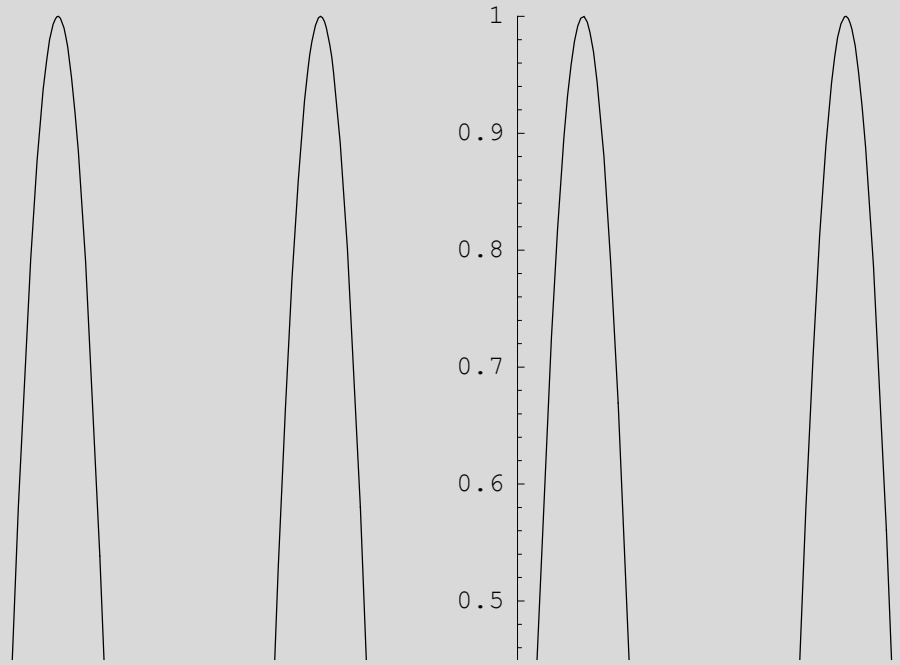
Άskhsh: **Allάxte tiV timέV tou** PlotRegion (**protimήsete timέV metax**ú 0 **kai** 1 **p.c** PlotRegion**Ø**{{0.5,1},{0,0.5}} **Deíte to apotélesma**. **Ajairéste to** ólo **to** PlotRegion **kai deíte to apotélesma**!
To {0.5,1} **sto** PlotRegion**Ø**{{0.5,1},{0,0.5}} **shmaínei óti to grάjhma qa katalάbei to diάsthma** 50% **ewV** 100% **sth orizóntia kateúqunsh tou plaisíou**(**dhl**. **to** ήmisu **dexi**ó) **en**ώ **to** {0,0.5} **shmaínei óti to grάjhma qa katalάbei to diάsthma** 0% **ewV** 50% **sth kάqeth kateúqunsh tou plaisíou**(**dhl**. **to kάtw** ήmisu)
Mporoúme kai na zoumάroume qétontaV pio mikró PlotRange. **To** PlotRange **écei scésh na kάnei me ta shmeia** (x,y) **thV kampúlhV**, **pou epiqumoúme na emjanízontai st**o **grάjhma**. To PlotRegion **écei scésh me thn telik**ή **qésh kai emjάnish tou grajήmatoV mésa sto plaísio**(**pou qa emjanízetai sthn oqónh maV ótan kάnoume klík sto grάjhma**)

```
Plot@Sin@3 xD, 8x, -3 Pi ê 2, 5 Pi<,
  PlotRange → 88-4.3, 4<, 80.45, 1<<, Axes -> 8False, True<D;
```



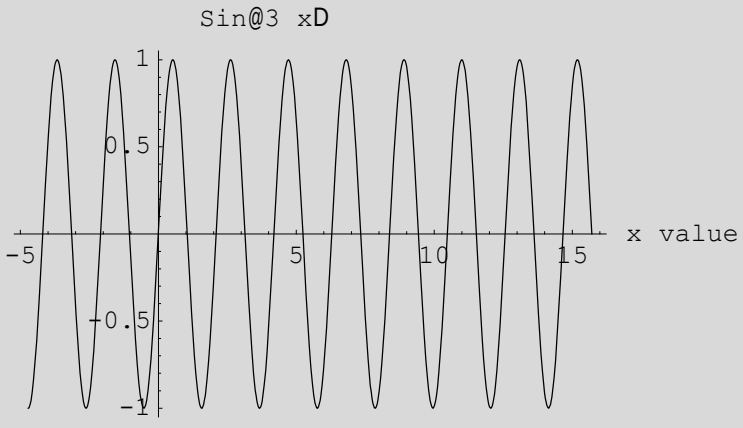
Gia na zoumάroume akóma perissótero mporoúme na trabήxoume proV ta éxw tiV labéV ή na qésoume mia megάlh timή sto ImageSize **p.c** ImageSize**Ø**400

```
Plot@Sin@3 xD, 8x, -3 Pi ê 2, 5 Pi<, PlotRange → 88-4.3, 4<, 80.45, 1<<,
  Axes -> 8False, True<, ImageSize → 400D;
```



στ) ΆxoneV. **Me** Axes->{False,True} **anagkάsame na mhn scediastεί o άxonaV** Ox. M**poroúme na bάloume etikέteV stouV άxoneV**

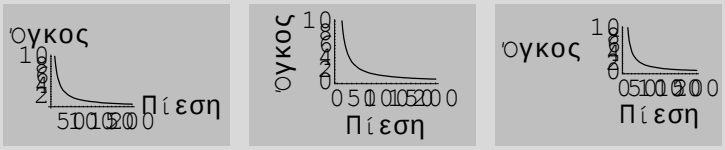
```
Plot@Sin@3 xD, 8x, -3 Pi ê 2, 5 Pi<,
 AxesLabel → 8"x value", "              Sin@3 xD"<D
```




```
¦ Graphics ¦
```

ζ) **Frames kai GridLines**. **Mporoúme na báloume plaísia gúrw apo thn kampúlh**. **H basikή entolή eínai** Frame->True. **Me** FrameStyle, FrameLabel, FrameTicks RotatedLabel **kai** GridLines **mporoúme na prosqésoume kápoia eidiaítera carakthristiká sto telikó plaísio**. **P.c**

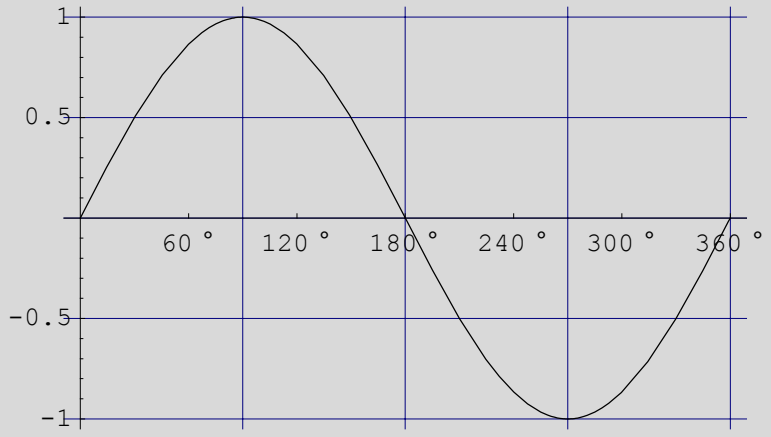
```
gr1 = Plot@100êP, 8P, 10, 200<, AxesLabel -> 8"Πίεση", "όγκος"<,
   Background -> GrayLevel@3ê4D, DisplayFunction -> IdentityD;
gr2 = Show@gr1, Frame -> 8True, True, False, False<,
   FrameLabel -> 8"Πίεση", "όγκος"<D;
gr3 = Show@gr2, RotateLabel -> FalseD;
Show@GraphicsArray@8gr1, gr2, gr3<DD;
```




```
Null[4]
```

(**Trabήxte proV ta éxw apó tiV labéV an creiasteí na megenqúnete**) **Me** Frame->{True,True,False,False} **bálame plaísio móno aristerá kai kátw**. **Me** RotateLabel->False **stríyame thn etikéta orizóntia me apotélesma na écoume ceiróterh emjánish sto grájhma** gr3 **ap' óti sto** gr2 **sto opoío h proepilegménh timή tou** RotateLabel **eínai** RotateLabel->True. **Me** GridLines **mporoúme na prosqésoume plégma sta shmeía pou qéloume** (**me proepilegménh timή** GridLines**Ø**Automatic, **to plégma topoqeteítai automata apo to** *Mathematica* **me kápoio eswterikó algóriqmo**) **p.c**

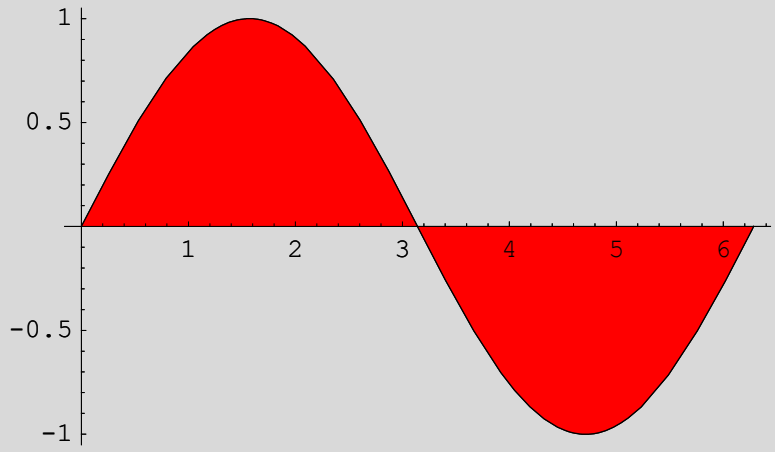
```
units = Range@0 Degree, 360 Degree, 60 DegreeD;
PlotASin@xD, 8x, 0, 2 Pi<, Ticks → 8units, Automatic<,
  GridLines → 99Pi/2, Pi, 3 Pi/2, 2 Pi, 5 Pi/2=, Automatic=E;
```



9.1.4 Άλλες δυνατότητες των δισδιάστατων γραφικών

Μπορούμε με FilledPlot του πακέτου Graphics`FilledPlot` **να γεμίσουμε με ένα χρώμα το χώρο που βρίσκεται μεταξύ δυο ή παραπάνω καμπυλών καμπυλών ή μεταξύ μιας καμπύλης και ένα άξονα**. **Έτσι για παράδειγμα η** FilledPlot[f[x],g[x],{x,xin,xmax}] **γεμίζει με χρώμα τον χώρο μεταξύ της** f[x], **και** g[x] **για** {x,xin,xmax}, **ενώ με σκέτο** FilledPlot[f[x],{x,xin,xmax}] **ή με** FilledPlot[f[x],0,{x,xin,xmax}] **γεμίζεται ο χώρος μεταξύ της** f[x] **και του Ox**. **Ρ.χ**

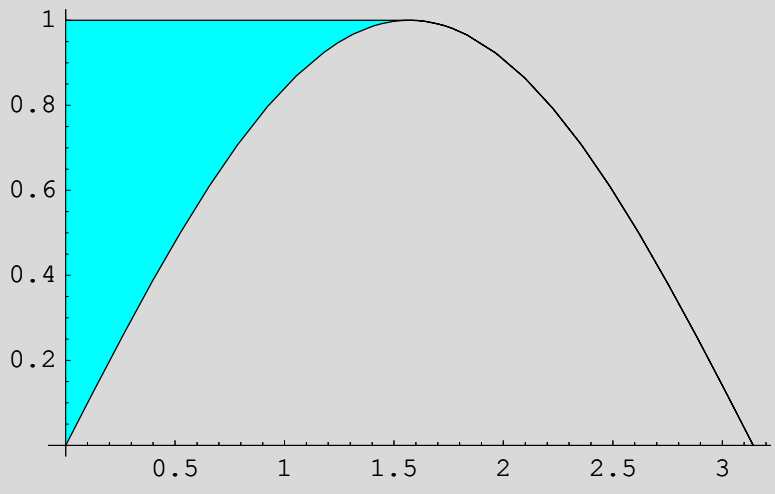
```
<< Graphics`FilledPlot`
FilledPlot@Sin@xD, 8x, 0, 2 Pi<, Fills → 8RGBColor@1, 0, 0D<D
```




```
¦ Graphics ¦
```

Στο επόμενο παράδειγμα βάλαμε χρώμα μεταξύ της f[x]=Sin[x] **και της ευθείας** g[x]=1 **και μόνο στο διάστημα** [0.Pi/2]. **Αναγκαστήκαμε να ορίσουμε την** g[x]=1 **για το διάστημα** [0,Pi/2] **και** f[x] **στο υπόλοιπο διάστημα έτσι ώστε να βγεί το επιθυμητό αποτέλεσμα**

```
FilledPlot[ { Sin[x], If[x<=Pi/2&&x>=0,1,Sin[x]]},
        {x, 0, Pi},  Curves -> Front];
```



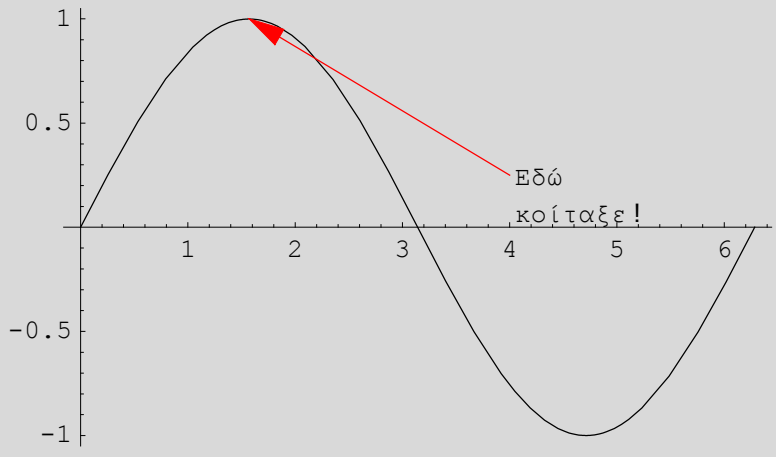
Me PlotVectorField tou pakétou Graphics`Plotfield` **paírnoume thn grajikh parástash enóV dianusmatikoú pedíou enó me** CartesianMap **kai** PolarMap **tou pakétou** Graphics`ComplexMap` , **scediázoume migadikéV sunartήseiV. Me PlotGradientField**[f[x,y],{x,xmin,xmax},{y,ymin,ymax}] **tou pakétou** Graphics`-PlotField` **paírnoume thn kateúqunsh kata thn opoía h sunárthsh maV** f[x,y] **auxánetai me ton tacútero ruqmó apo to shmeío** {x,y}. **Den prépei na xecásoume kai thn Animate tou pakétou** Graphics`Animation` **pou maV dínei thn dunatóthta thV kinoúmenhV eikónaV lísteV shmeíwn ή lísta sunartήsewn. Parakátw blépoume thn kataskeuή miaV akolouqíaV órwn crhsimopoiόntaV thn paráδeigma** ListPlot, Arrow <<Graphics`Arrow`,

```
<< Graphics`Animation`
<< Graphics`Graphics`
AnimateAListPlotATableA 1/n + SinA n π/2 E + CosA n π/2 E, 8n, m<E,
  PlotRange → 880, 20<, 8-1.5, 1.5<<, AxesOrigin → 80, 0<,
  PlotStyle → 8PointSize@0.02D, Hue@.6D<E, 8m, 1, 20, 1<E
```

PatώntaV diplό klík se éna apo ta parapánw grajήmata écoume thn zhtoúmenh Animation. **Prosέxte kalέsame dúo diajoretiká pakéta gia dúo diajoretikoúV skopoúV. Kalέsame to** Graphics`Graphics` **gia thn** ListPlot **kai to** <<Graphics`Animation` **gia to** Animate. **Apo thn kataskeuή lépoume óti h akolouqía maV** 1/n + Sin@n πD + Cos@n πD **écei duo sugklínouseV upakolouqíeV**.

Τελος με **Arrow[{x0,y0},{x1,y1}]** μπορούμε να σχεδιάσουμε ένα βέλος απο το σημείο A_0 Hx0, y0Lστο A_1 Hx1, y1L. Είναι χρήσιμο όταν θέλουμε να σχεδιάσουμε βέλη ή προσανατολισμένους άξονες ή όταν θέλουμε να τονίσουμε ένα τμήμα του γραφήματος. Η Arrow απαιτει το πακέτο Graphics`Arrow` π.χ με την βοήθεια

```
<< Graphics`Arrow`
Plot@Sin@xD, 8x, 0, 2 Pi<,
  Epilog → 88RGBColor@1, 0, 0D, Arrow@84, .25<, 8Pi ê 2, 1<D<,
    Text@"Εδώ \nκοίταξε!", 84, .15<, 8-1, 0<D<D;
```



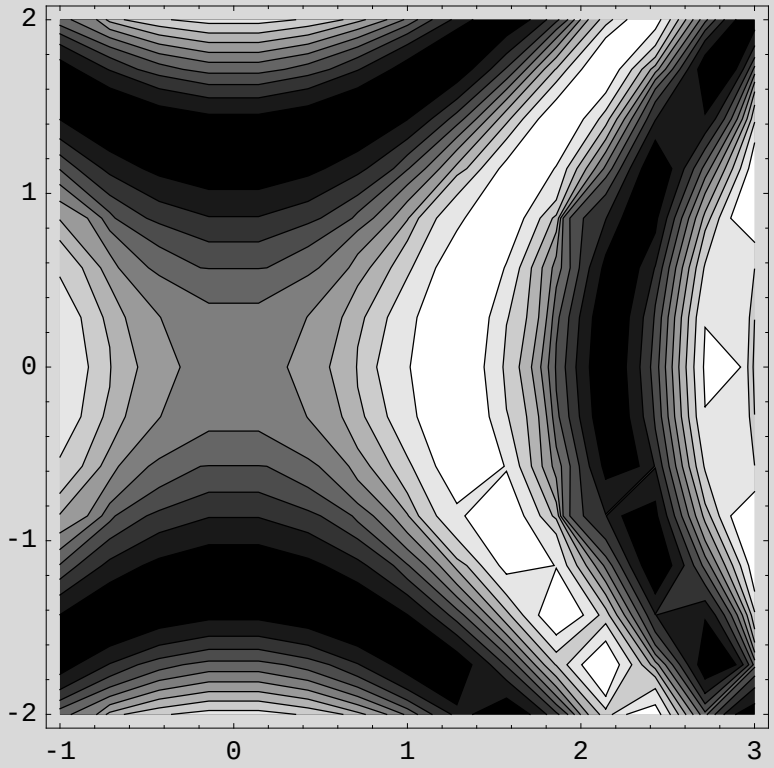
Με το \n μπορούμε να αλλάξουμε γραμμή στο κείμενό μας!

9.2 Μελετώντας τρισδιάστατα γραφικά στο επίπεδο

9.2.1 Oi sunarthseiV Contour Plot kai DensityPlot

Me thn ContourPlot[f[x,y], {x,xmin,xmax},{y,ymin,ymax}] **scediázoume thn** f[x,y] **pánw sto epípedo** Oxy, **dínontaV sto shmeío** (x,y) **éna crώma** (**sunήqwV apócrwsh tou gkrízou**) **pou antistoiceí sthn timή** f[x,y]. **Ta shmeía pou écoun megalúterh timή** f[x,y] **eínai pio jwteinά enώ autά pou écoun mikróterh eínai pio skoteinά**. **P.c**

```
ContourPlot@Sin@x^2 - y^2D, 8x, -1, 3<, 8y, -2, 2<D;
```

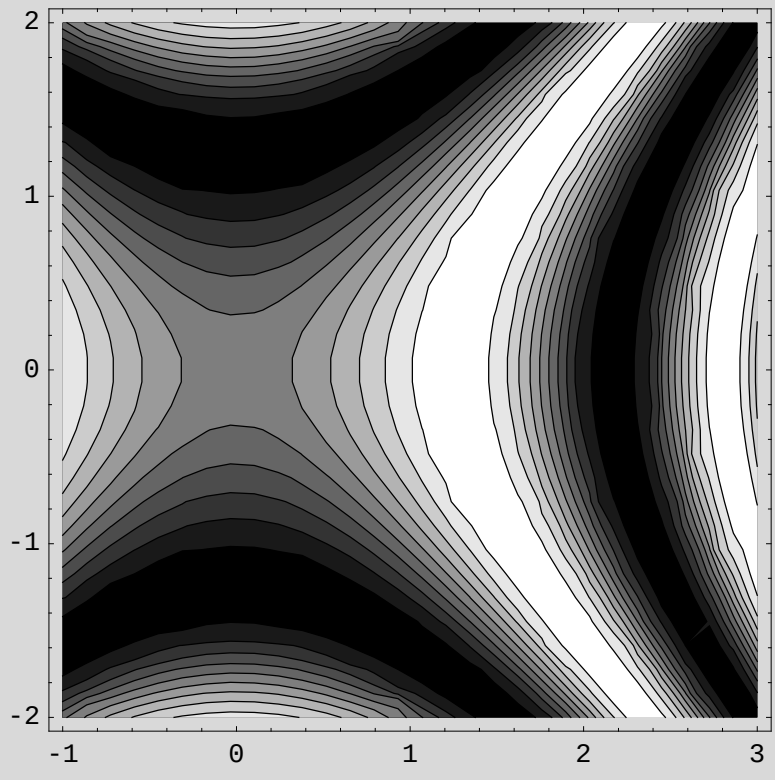


Parathroúme 10 **apocrώseiV tou grí**(**to leukó den to metráme**, **ousiastikά écoume mazí me to leukó** 11). **H na to poúme diajoretikά**: **To pedío timών**(**ston άxona O**z) **écei cwristeí se** 10+1 **ísou mήkouV diastήmata éstw D**1,**D**2,..,**D**11. **Kάqe diάsthma paírnei éna crώma tou gkrí xekinώntaV apo to maúro**. **Όsa shmeía** (x,y) **tou epipédou apeikonízontai**(**mésw thV** f) **mésa sto ídio diάsthma** D_i **qa pároun thn ídia apócrwsh**! **'Etsi pánw sto epípedo O**xy **emjanízontai crwmatikéV lwrídeV**,**ta isouyή epípeda**, **ta opoía diacwrízontai metaxú touV apo kάpoieV kampúleV pou légontai isouyeíV**(Contours). **Όla ta shmeía miaV isouyή kampúlhV paírnoun thn ídia akribώV timή me thn** f ! **Fusikά epeidή upárcoun άpeira shmeía** (x,y) **mésa sto orqogώnio scediasmoú** D={x,xmin,xmax}X{y,ymin,ymax}, **to** *Mathematica* **qa dialέxei deigmatoleiptikά líga shmeía apo to** D **kai me bάsh tiV timέV touV qa scediάsei tiV isouyeíV kampúleV**. **Ta shmeía autά légontai** Plot-Points. **Fusikά to apotélesma pou paírnoume écei ópwV blépoume pollέV atéleieV**. **Gia parάdeigma oi isouyeíV kampúleV den eínai óso qa perimέname omalέV**. **EpishV mporoúme na parathrήsoume óti h leukή lwrída sta dexiά eínai sto kάtw méroV thV kommatiasménh**! **Autó wjeíletai kuríwV sto gegonόV óti h proepilegménh timή tou** PlotPoints **eínai** 15. **Opóte apo to diάsthma** D **epilégontai** 15X15 **to plήqoV shmeía pou den eínai arketά an h** f[x,y] **écei apótomeV "lakoúbeV" kai "lojískouV" sto** D. **Qa prospaqήsoume na antimetwpísoume autέV tiV atéleieV**. **AV doúme ómwV prώta**, **poiέV eínai oi epilogέV thV** ContourPlot:

```
Options@ContourPlotD

8AspectRatio → 1, Axes → False, AxesLabel → None,
 AxesOrigin → Automatic, AxesStyle → Automatic,
 Background → Automatic, ColorFunction → Automatic,
 ColorFunctionScaling → True, ColorOutput → Automatic,
 Compiled → True, ContourLines → True, Contours → 10,
 ContourShading → True, ContourSmoothing → True,
 ContourStyle → Automatic, DefaultColor → Automatic, Epilog → 8<,
 Frame → True, FrameLabel → None, FrameStyle → Automatic,
 FrameTicks → Automatic, ImageSize → Automatic, PlotLabel → None,
 PlotPoints → 15, PlotRange → Automatic, PlotRegion → Automatic,
 Prolog → 8<, RotateLabel → True, Ticks → Automatic,
 DefaultFont Ç $DefaultFont, DisplayFunction Ç $DisplayFunction,
 FormatType Ç $FormatType, TextStyle Ç $TextStyle<
```

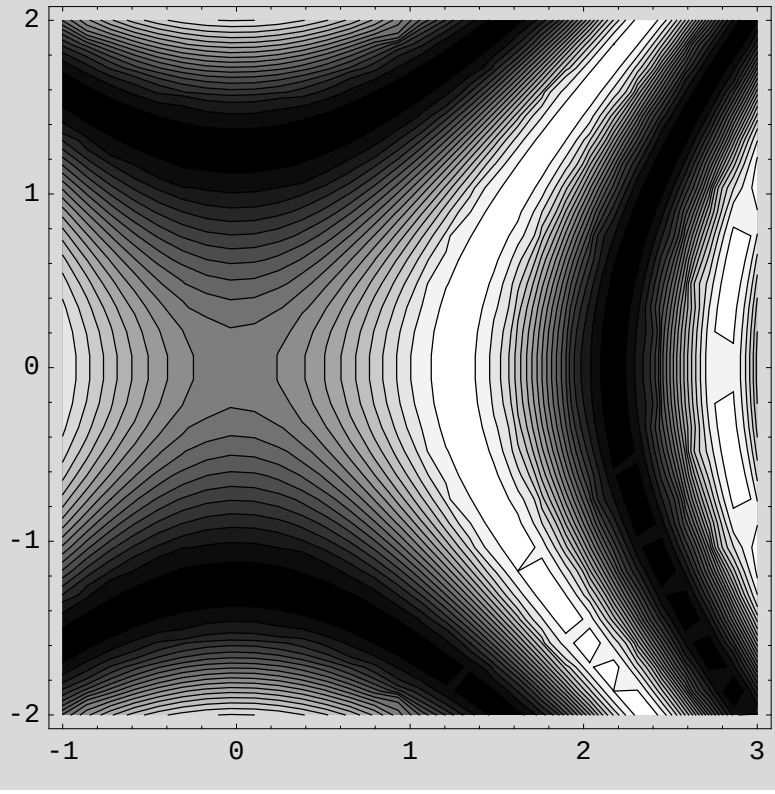
AllázontaV **kápoia apo ta parapánw carakthristiká mporoúme na écoume éna kaló apotélesma**. **P**.**c mporoúme na epitréyoume sto** *Mathematica* **na kánei kalúterh deigmatoleiyía paírnonta**V **perissótera shmeía**. **WV apotélesma qa écoume pio akribeíV isouyeíV kampúleV**. **P**.**c**

```
gr = ContourPlot@Sin@x^2 - y^2D,
   8x, -1, 3<, 8y, -2, 2<, PlotPoints → 30D;
```



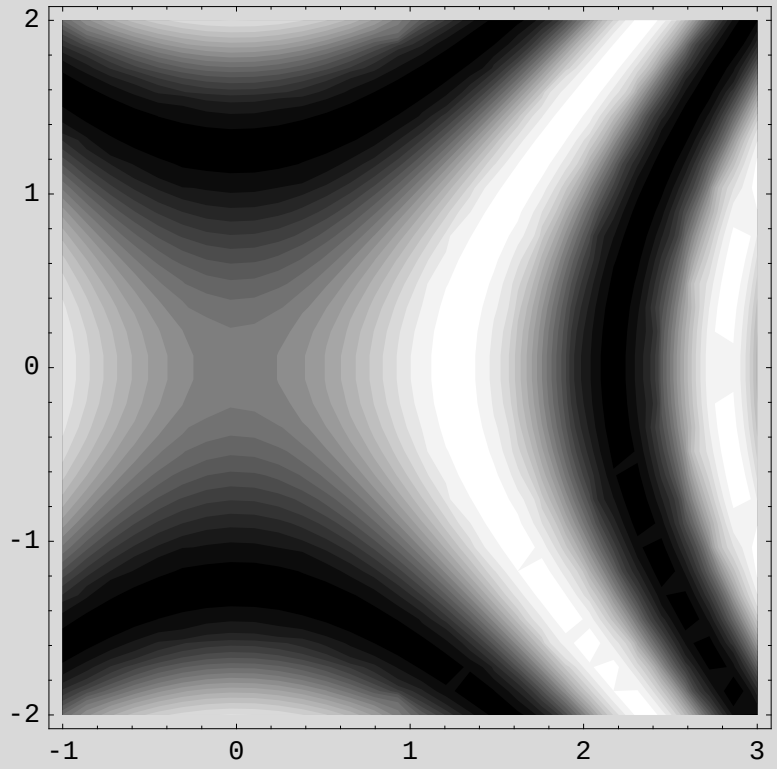
Mporoúme epíshV na zhtήsoume perissótereV isouyeíV(**kai ára perissótereV apocrώseiV**) **megalώnontaV thn proepilegménh timή tou** Contours **pou eínai** 10(ή **akribéstera** 10+1 ópwV **proanajérame**). **P**.**c**

```
gr1 = Show@gr, Contours → 20D;
```



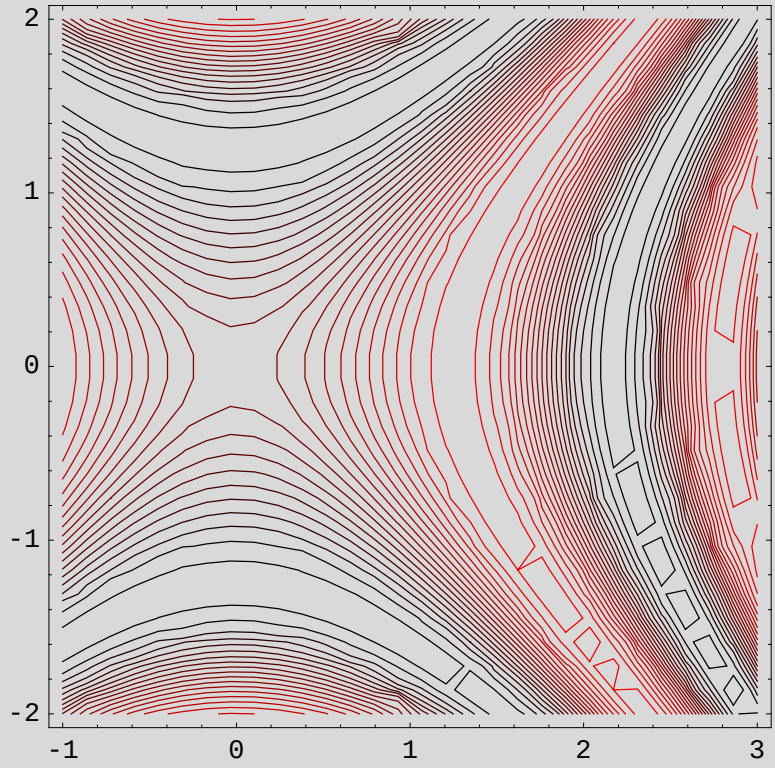
Prosέxte óti auxánontaV to plήqoV twn Contours **cánetai h akríbeia sto scediasmó twn isouyών**!!! Ára **qa prépei na auxήsoume xaná ta** PlotPoints **gia na petúcoume thn akríbeia ston scediasmó**! **An tώra qéloume móno tiV diabaqmíseiV cwríV tiV isouyeíV qa grájame** ContourLines->False **p.c**

```
gr2 = Show@gr1, ContourLines → FalseD;
```



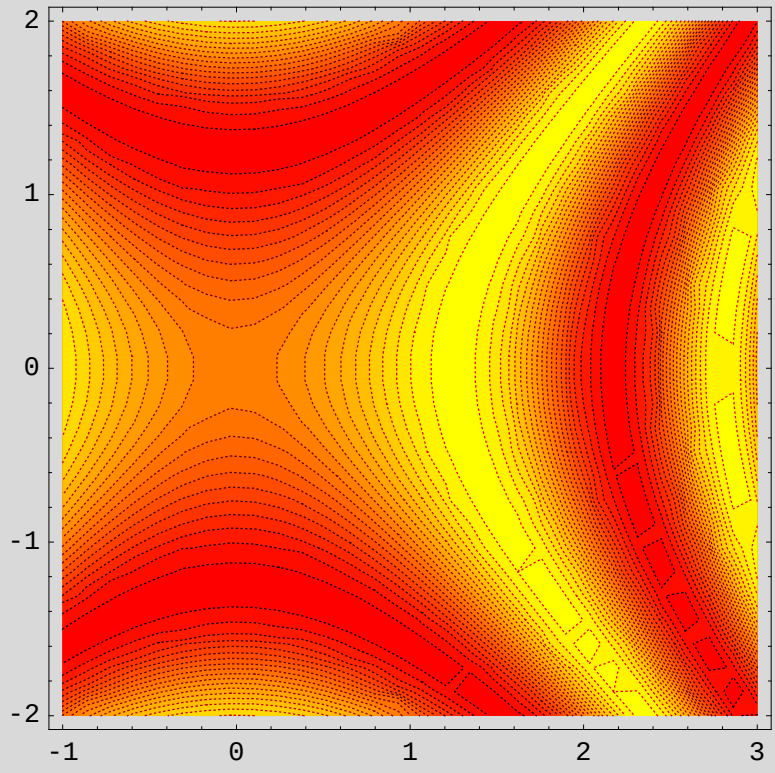
Sígoura polú pio katanohtó apotélesma! MerikéV joréV den maV endiajéroun tóso oi apocrώseiV óso oi ídieV oi isouyeíV! Parakátw dínoume éna tétoio paráδeigma. Me ContourShading->False exajanízoume tiV apocώseiV enώ me ContourStyle->({RGBColor[#,0,0]}&)/@Range[0,0.95,.05] dínoume 20 **diajoretikéV apocrώseiV(ósa eínai kai ta** Contours) **tou kókkinou stiV antístoiceV isouyeíV kampúleV.**

```
gr3 = Show@gr1, ContourShading -> False,
  ContourStyle → H8RGBColor@#, 0, 0D< &L ê@ Range@0, 0.95, .05DD;
```



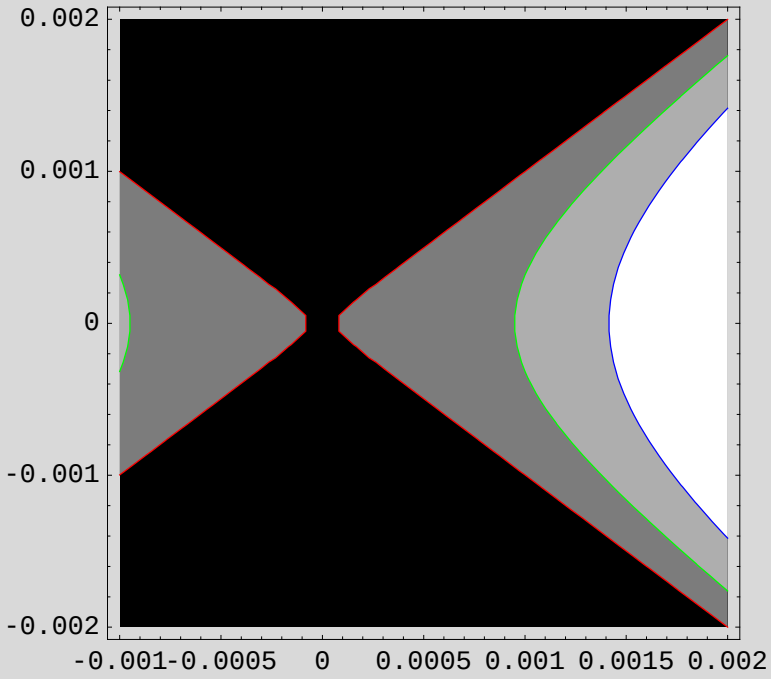
Edώ me éntono kókkino eínai oi isouyeíV pou brískontai pio yhlá apo tiV álleV. Qa mporoúsame tώra na emjanísoume kai ta isouyή epípeda me diabaqmíseiV tou kítrinou-kókkinou(me thn boήqeia thV Color-FunctionØ(RGBColor[1,#,0]&)) kai tiV ContourLines kókkineV kai diakekommέneV p.c

```
gr4 = Show@gr1, ContourShading → True,
  ColorFunction → HRGBColor@1, #, 0D &L,
  ContourStyle → H8RGBColor@#, 0, 0D, Dashing@80.0015, 0.005<D< &L ê@
    Range@0, 0.95, .05DD;
```



Den prépei na xecásoume na anajéroume thn Contours->{z1,z2,z3,...} me thn opoía epilégoume na mpoún isouyeíV móno stiV sugkekrimméneV timéV tou z. P.c qa prospaqήsoume na diereunísoume thn f kontá sto shmeío (0,0) meletώntaV móno kápoieV qetikéV isouyeíV me timéV kontá sto f[x,y]=0 p.c :ContoursØ84 10^{-9}, 9 10^{-7}, 2 10^{-6}<Gia akríbeia megalώnoume kai to plήqoV twn deigmatoleiptikών shmeíwn(PlotPointsØ40)

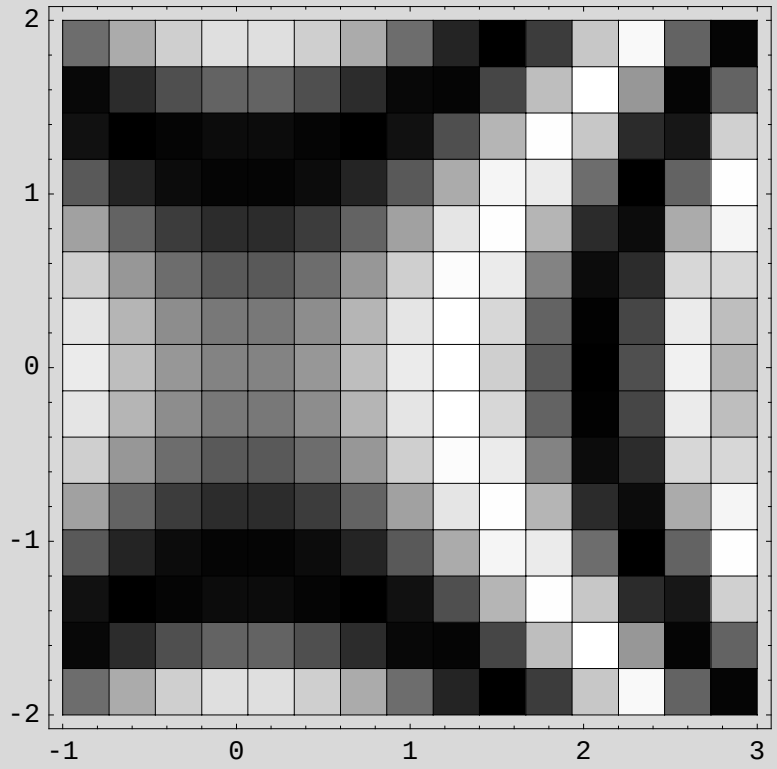
```
gr5 =
  ContourPlot@Sin@x^2 - y^2D, 8x, -0.001, 0.002<, 8y, -0.002, 0.002<,
   Contours → 84 10^-9, 9 10^-7, 2 10^-6<, PlotPoints → 40, ContourStyle →
    88RGBColor@1, 0, 0D<, 8RGBColor@0, 1, 0D<, 8RGBColor@0, 0, 1D<<D;
```



Me maúro crώma eínai óleV oi timéV thV sunárthshV $<4\,10^{-9}$, **me endiámeso gkrí oi timéV metaxú** $4\,10^{-9}$ και $9\,10^{-7}$ **k.o.k**

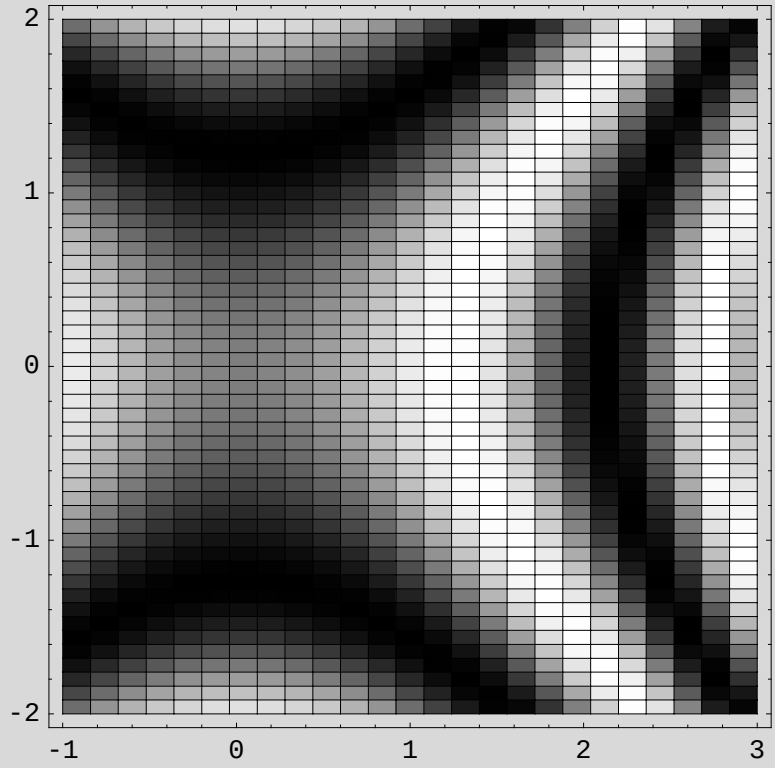
H DensityPlot den prospaqeí na scediásei kápoieV isouyeíV kampúleV ópwV h ContourPlot. **AplώV parágei éna plégma**(mesh)**kai kápoieV apocrώseiV mésa se autó. H proepilegméneV apocrώseiV eínai tou gkrí. Skoteinά gkrí crhsimopoioúntai gia baqoulώmata thV** f[x,y] **dhl. gia mikréV timéV kai anoikta gkrí gia megáleV timéV. P.c**

```
DensityPlot@Sin@x^2 - y^2D, 8x, -1, 3<, 8y, -2, 2<D;
```



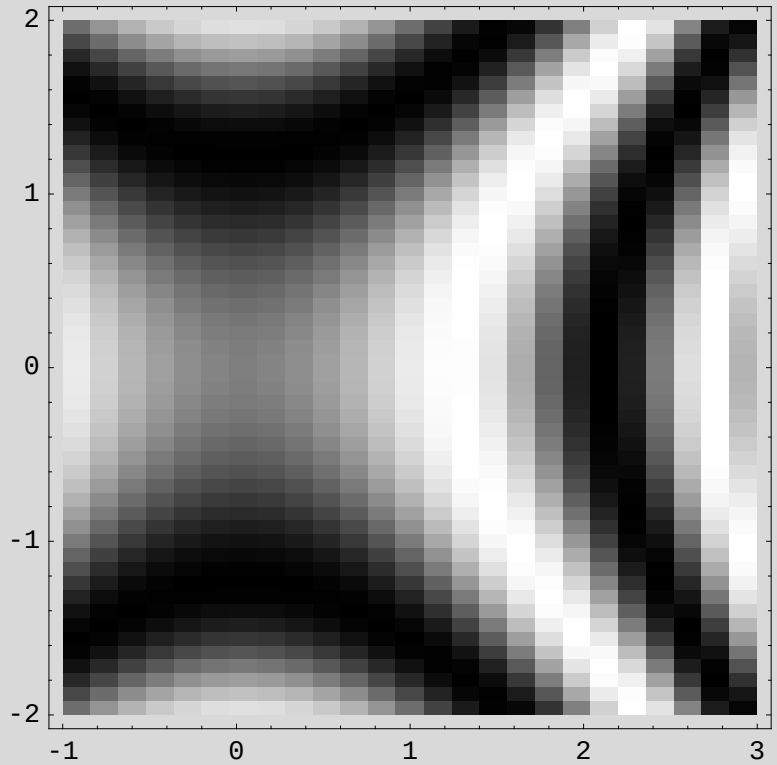
Όpwv blέpoume crhsimopoioύntai 15 PlotPoints se kάqe éna apo ta diastήmata twn x kai y antίstoica. Gia megalύterh akríbeia sta crώmata mporoύme na bάloume megalύterh timή p.c PlotPoints->{25,50}

```
gr1 = DensityPlot@Sin@x^2 - y^2D,
   8x, -1, 3<, 8y, -2, 2<, PlotPoints → 825, 50<D;
```



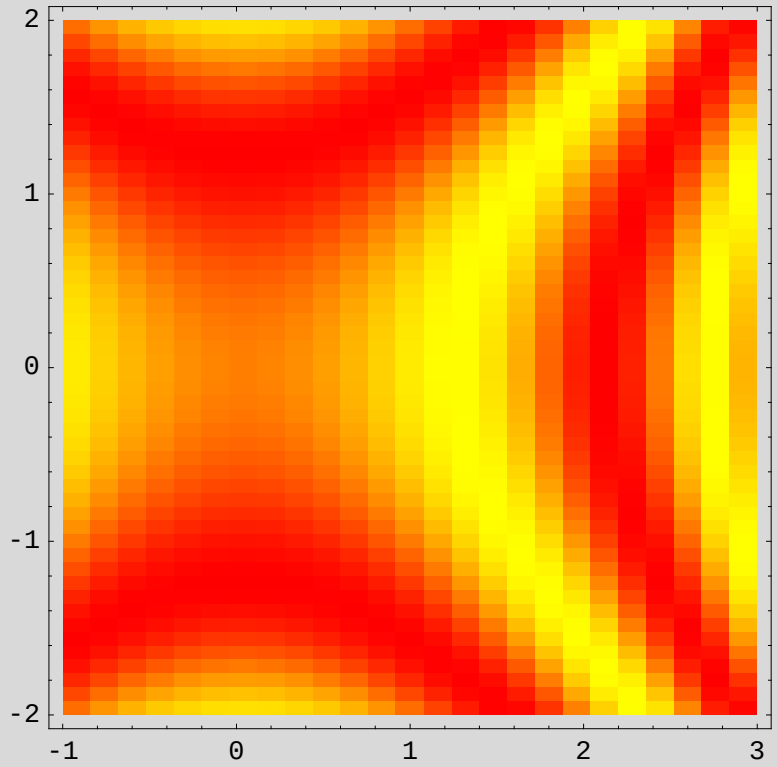
Όpwv blέpoume den gínetai kammía prospáqeia na schmatistoún kapoia isouyή epípeda AplώV se káqe deigmatolhptikó shmeío apo ta 25 **ä**50 **upologízetai h antístoich timή thV** f **kai sthn sunéceia autή metatrépetai se mia apócrwsh tou gkrí**. **Me** Mesh->False **mporoúme na exajanísoume to plégma kai na meínei móno h apócrwsh**:

```
gr2 = Show@gr1, Mesh → FalseD;
```



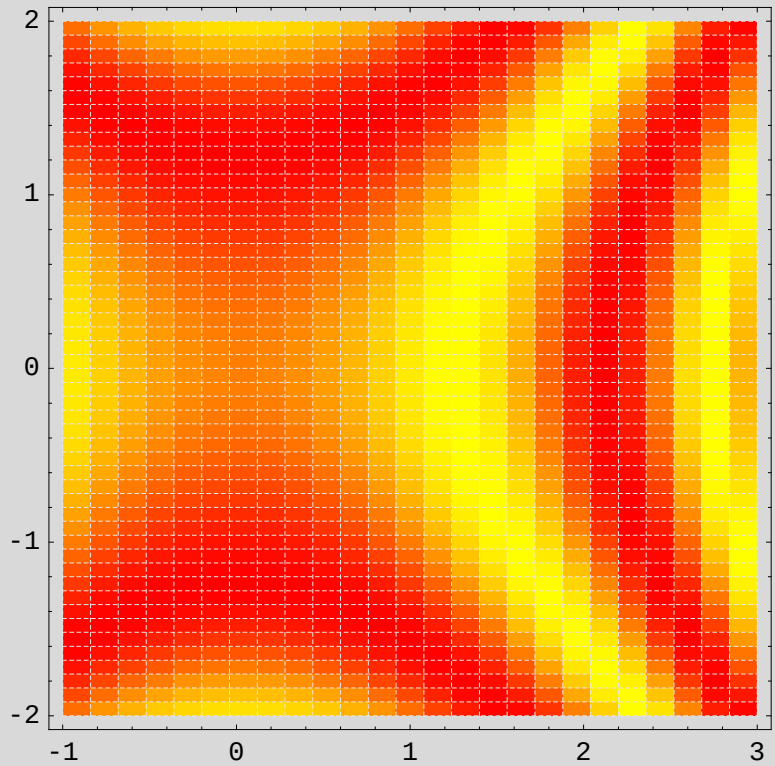
Με την ColorFunction μπορούμε να αλλάξουμε κατά βούληση τις αποχρώσεις:

```
gr3 = Show@gr2, ColorFunction → HRGBColor@1, #, 0D &LD;
```



An qéloume na emjanízontai oposdήpote kai to plégma qa ήtan skópimo na dialégame me thn boήqeia thV MeshStyle **éna diajoretikó croώma grammώn plégmatoV ή pio leptéV grammέV plégmatoV ή diakekommέneV ή kάpoia apo ta prohgoύmena**:

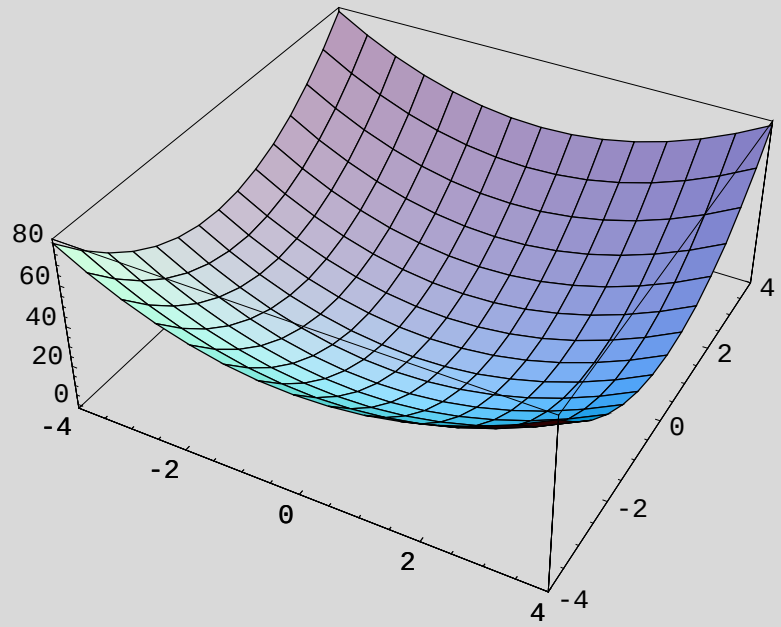
```
gr4 =
  Show@gr1, MeshStyle → 8GrayLevel@0.9D, Dashing@80.005, 0.005<D<,
   ColorFunction → HRGBColor@1, #, 0D &LD;
```



Genikά h DensityPlot **maV scediάzei mia epijάneia tou cώrou όpwV qa thn έblepe énaV parathrhtήV pou briskόtan akribώV apo pάnw thV**!
Ίswς qa anarwtiέste giatί na crhsimopoiήsoume thn DensityPlot **ajoύ upάrcei h** ContourPlot. **H apάnthsh eínai óti upάrcoun kakέV periptώseiV pou h** ContourPlot **sthn prospάqeia thV na zwgrajίsei ta isouyή epίpeda den bgάzei kάpoia kάpoio** κατανοητό **grάjhma dhl**. **maV epistréjei anakribέV grάjhma**. **Genikά qa prépei na eímaste se qέsh na paírnoume όleV tiV plhrojoríeV pou maV creiάzontαι sthn melέth maV kάnontaV katάllhlo sunduasmό ólwn twn dunatotήtwn p.c**

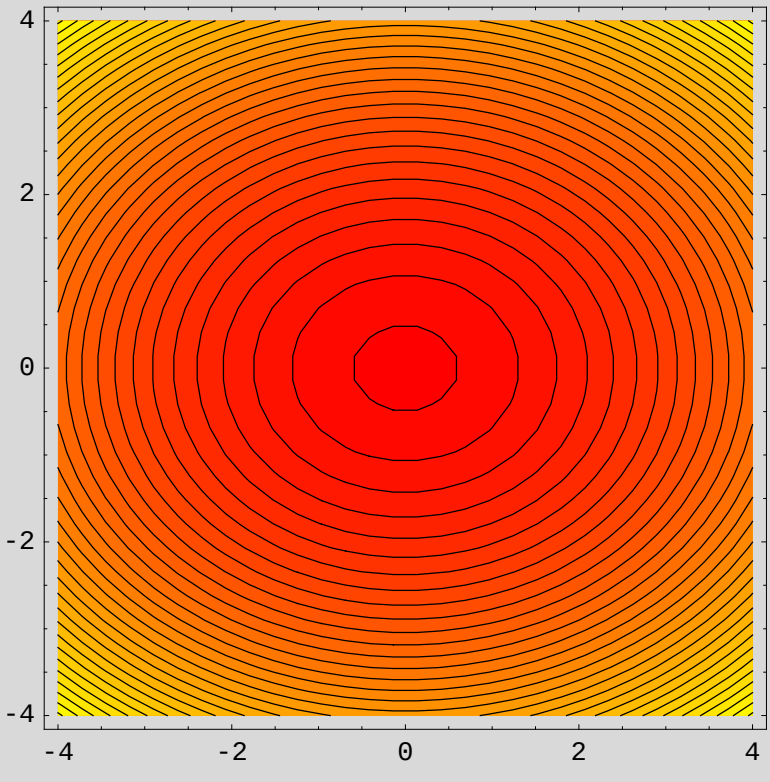
Askhsh: Na meletήsoume thn sumperijorά thV 2 x^2 + 3 y^2 **gia** {x,-4,4} **kai** {y,-4,4}.
Lύsh:**Crhsimopoioύme thn** Plot3D **se sunduasmό me thn** ContourPlot:

```
Plot3D@2 x^2 + 3 y^2 , 8x, −4, 4< , 8y, −4, 4<D;
```



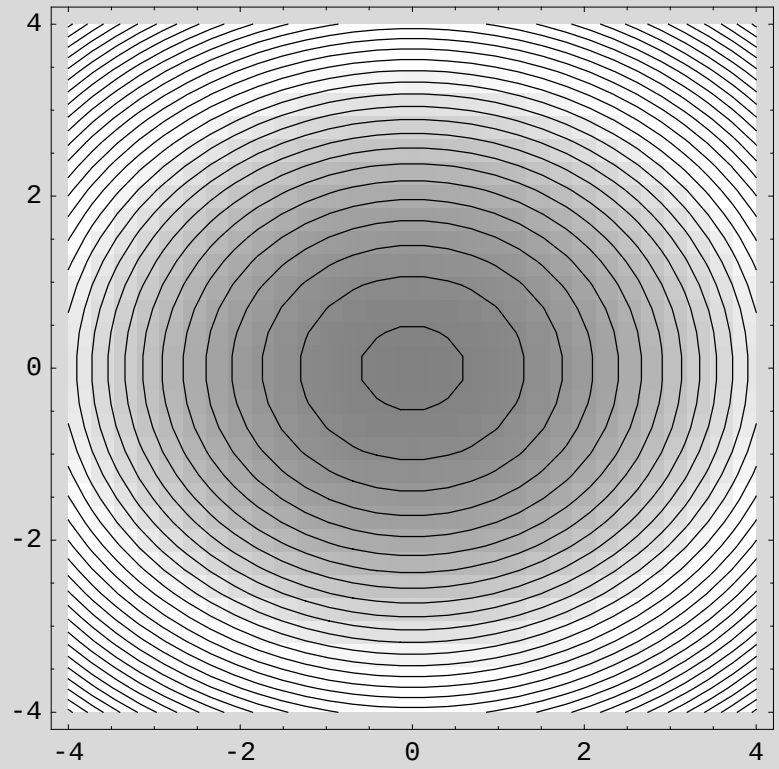
Blépoume óti upárcei éna baqoúlwma allá den xéroume pou akribóV. H ContourPlot **qa bohqήsei ston entopismó tou**:

```
ContourPlot@2 x^2 + 3 y^2 , 8x, −4, 4< ,
  8y, −4, 4<, ColorFunction → HRGBColor@1, #, 0D &L,
  Contours → 30, PlotPoints → 30D;
```



Dhl. **eínai to shmeío** (0,0)! **Qa mporoúsame na crhsimopoíhsoume kai thn** ContourPlot **se sunduasmó me thn** DensityPlot **wV exήV**: Θέτουμε την DensityPlot κάτω απο την ContourPlot και στην πρώτη βάζουμε Mesh→False ενώ στην δεύτερη ContourShading→False(για να εμφανιστούν μόνο οι ισουψείς καμπύλες)

```
cg = ContourPlot@2 x^2 + 3 y^2 ,
   8x, −4, 4< , 8y, −4, 4<, ContourShading → False,
   DisplayFunction → Identity, PlotPoints → 30, Contours → 30D;
dg = DensityPlot@2 x^2 + 3 y^2 , 8x, −4, 4< , 8y, −4, 4<,
   PlotPoints → 30, Mesh → False, DisplayFunction → Identity,
   ColorFunction → HGrayLevel@If@0.5 + # > 1, 1, 0.5 + #DD &LD;
Show@dg, cg, DisplayFunction → $DisplayFunctionD;
```



Me If[0.5+#>1,1,0.5+#] **jwtísame katá** 0.5 **perissótero ta skoteinά merh tou** DensityPlot **gia na écoume perissóterh jwteinóthta sta shmeía gúrw apo to** (0,0).

9.3 Τρισδιάστατες Γραφικές Παραστάσεις

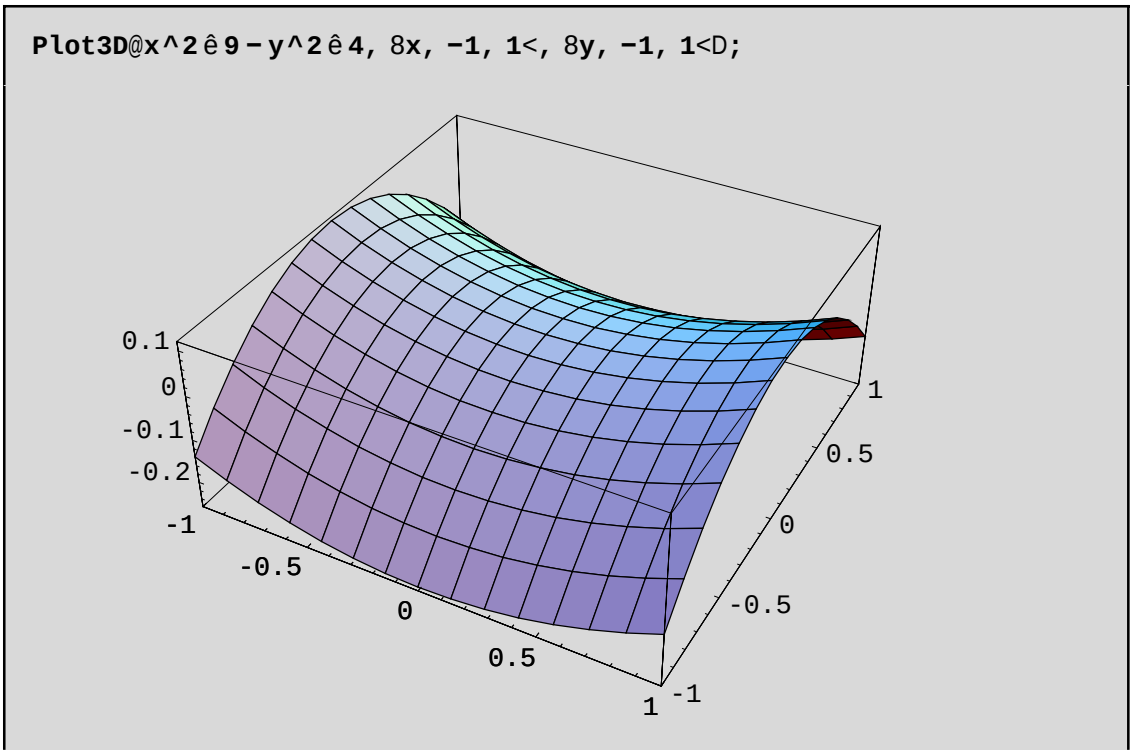
Υπάρχει αρκετή ομοιότητα στις εντολές και στις επιλογές για την γραφική παράσταση επιφανειών και καμπυλών στο χώρο, με τις αντίστοιχες που γνωρίσαμε στα διδιάστατα γραφικά. Έτσι σε πολλές περιπτώσεις δεν θα αναφέρουμε πολλές λεπτομέρειες. Γι'αυτό θα συνιστούσαμε να διαβάσετε ξανά το κεφάλαιο για τις διδιάστατες γραφικές παραστάσεις, και να ανακαλύψετε τις ομοιότητες και τις διαφορές που υπάρχουν μεταξύ των εντολών.

9.3.1 Γραφικές παραστάσεις επιφανειών.

Χρησιμοποιούμε διαφορετικές εντολές ανάλογα με τον τρόπο που περιγράφεται η επιφάνεια που μελετάμε. Έτσι έχουμε τις παρακάτω περιπτώσεις:

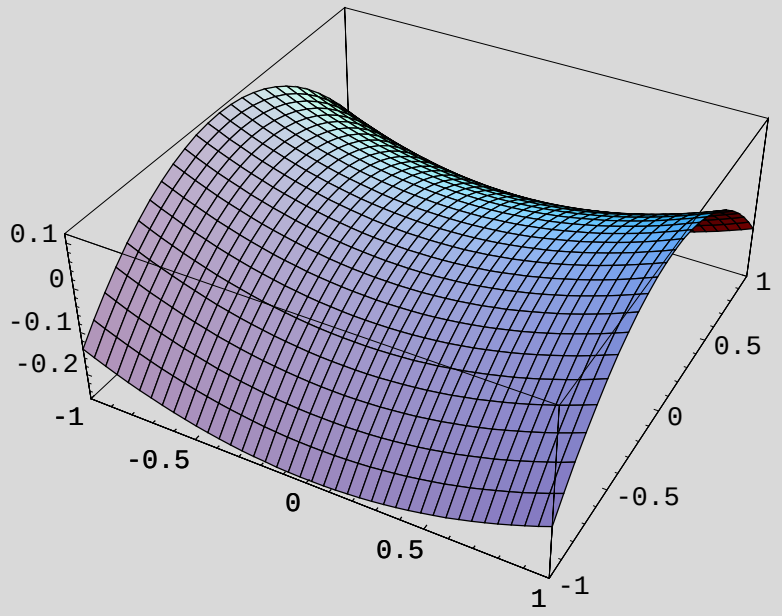
1. Η επιφάνεια είναι το **γράφημα μιας συνάρτησης** δυο μεταβλητών(π.χ της x και y). Τότε χρησιμοποιούμε την Plot3Dz π.χ εαν $f[x, y] = \frac{x^2}{9} - \frac{y^2}{4}$ και πεδίο ορισμού το [-1,1]X[-1,1] τότε παίρνουμε την εντολή

```
Plot3D[x^2/9 - y^2/4, {x, -1, 1}, {y, -1, 1}];
```

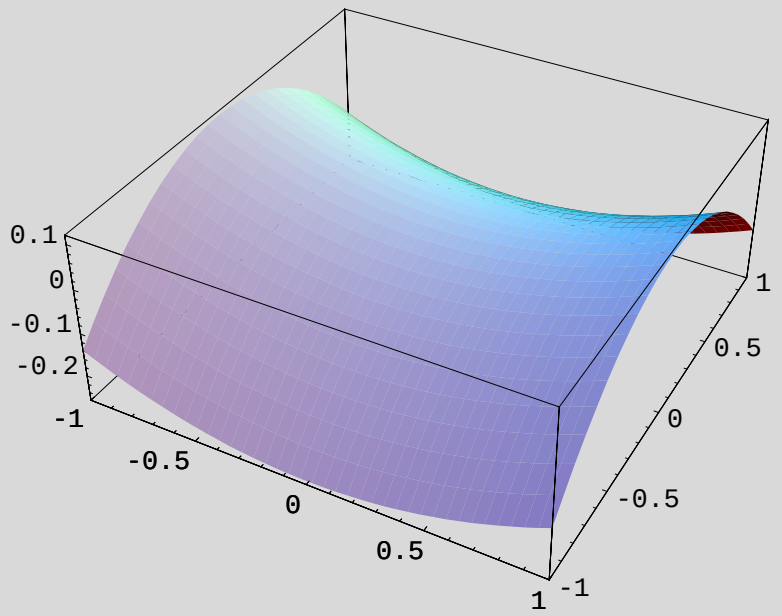


Η επιφάνεια αυτή δεν είναι άλλη απο το υπερβολικό παραβολοειδές δηλ. το "σαμάρι". Παρατηρούμε ότι η επιφάνεια έχει καλυφθεί απο καμπύλες παράλληλες με τους άξονες Ox και Oy. Αυτές τέμνονται κάθετα και έτσι σχηματίζεται ένα πλέγμα(Mesh). Ουσιαστικά επιλέγονται 15 σημεία σε κάθε κατεύθυνση οπότε έχουμε 15X15 σημεία πάνω στο επίπεδο Oxy! Αυτά είναι τα δειγματοληπτικά σημεία(PlotPoints) δηλ. σε κάθε ένα απο αυτά υπολογίζεται η τιμή της συνάρτησης f[x,y] και στην συνέχεια με βάση αυτές τις τιμές, σχεδιάζεται το πλέγμα και τελικά η ίδια η επιφάνεια . Είναι λοιπόν ευνόητο αν θέλουμε καλύτερη ακρίβεια στην γραφική παράσταση να πάρουμε περισσότερα PlotPoints π.χ με PlotPoint->{40,25}διαλέγουμε 40 σημεία (απο το πεδίου ορισμού της f) πάνω στο Ox και 25 πάνω στον Oy(πάλι απο το πεδίου ορισμού της f).Με Mesh->False σχεδιάζεται η επιφάνεια μας χωρίς να είναι εμφανές το πλέγμα πάνω σ'αυτήν. Π.χ

```
Plot3D@x^2ê9 - y^2ê4, 8x, -1, 1<, 8y, -1, 1<, PlotPoints → 840, 25<D;
```

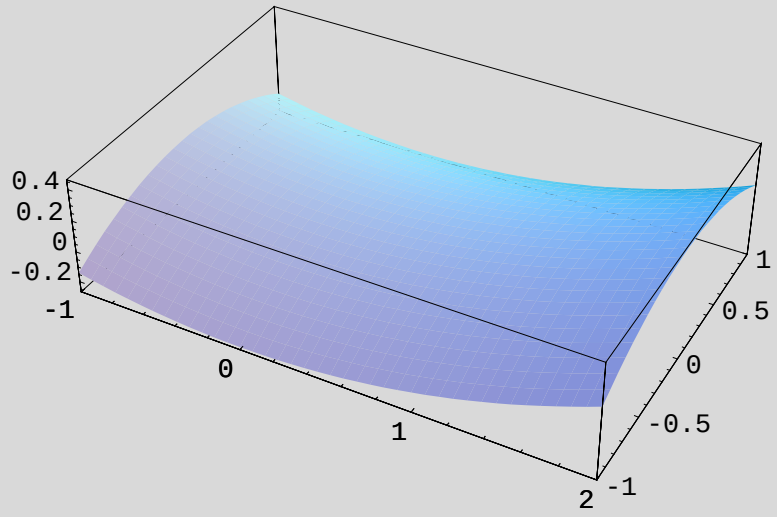



```
Plot3D@x^2ê9 - y^2ê4, 8x, -1, 1<,
  8y, -1, 1<, PlotPoints → 840, 25<, Mesh → FalseD;
```



Το διάφανο κουτί(Box) που περιβάλλει το πλέγμα, έχει λόγο πλευρών {1,1,0.4} δηλ. το μήκος του μήκους και του πλάτους είναι το ίδιο ενώ το ύψος του κουτιού είναι μικρότερο κατά 0.4 δηλ. κατα 40%. Αυτή είναι η προεπιλεγμένη τιμή του BoxRatios και μπορείτε να την αλλάξετε! Π.χ να θέσετε BoxRatios->{1,1,1} για να έχουν όλες οι πλευρές ίσο μήκος ή να θέσετε BoxRatios->Automatic π.χ

```
Plot3D@x^2ê9 - y^2ê4, 8x, -1, 2<, 8y, -1, 1<,
  PlotPoints → 840, 25<, Mesh → False, BoxRatios → AutomaticD;
```



Με Boxratios->Automatic έχουμε ίσες μονάδες μέτρησης σε κάθε άξονα και άρα το "φυσιολόγικό" γράφημα της επιφάνειας. Αυτό βέβαια δεν είναι πάντα επιθυμητό γιατί χάνουμε κάποια χαρακτηριστικά της επιφάνειας. Π.χ δεν φαίνεται καθαρά το "σαμαρωτό" σχήμα της επιφάνειας. Με BoxRatios->{1,1,2.5} θα έχετε καλύτερο αποτέλεσμα, δοκιμάστε!

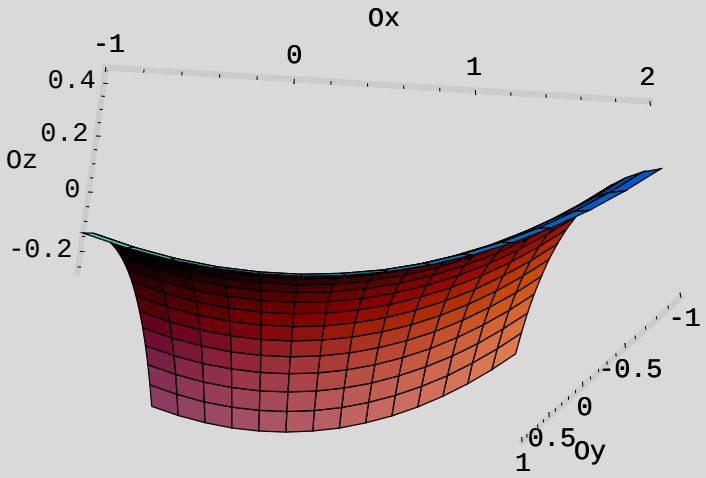
Μια πολύ χρήσιμη εντολή είναι η **Options[..]** διότι μπορούμε να διαπιστώσουμε τις επιλογές μιας εντολής καθώς και τις προεπιλεγμένες τιμές των επιλογών π.χ

```
Options@ParametricPlot3DD

8AmbientLight → GrayLevel@0.D, AspectRatio → Automatic,
 Axes → True, AxesEdge → Automatic, AxesLabel → None,
 AxesStyle → Automatic, Background → Automatic, Boxed → True,
 BoxRatios → Automatic, BoxStyle → Automatic, ColorOutput → Automatic,
 Compiled → True, DefaultColor → Automatic, Epilog → 8<,
 FaceGrids → None, ImageSize → Automatic, Lighting → True,
 LightSources → 8881., 0., 1.<, RGBColor@1, 0, 0D<, 881., 1., 1.<,
    RGBColor@0, 1, 0D<, 880., 1., 1.<, RGBColor@0, 0, 1D<<,
 Plot3Matrix → Automatic, PlotLabel → None, PlotPoints → Automatic,
 PlotRange → Automatic, PlotRegion → Automatic,
 PolygonIntersections → True, Prolog → 8<, RenderAll → True,
 Shading → True, SphericalRegion → False, Ticks → Automatic,
 ViewCenter → Automatic, ViewPoint → 81.3, -2.4, 2.<,
 ViewVertical → 80., 0., 1.<, DefaultFont Ç $DefaultFont,
 DisplayFunction Ç $DisplayFunction,
 FormatType Ç $FormatType, TextStyle Ç $TextStyle<
```

Παρατηρούμε ότι έχουμε πάρα πολλές δυνατότητες. Βέβαια αρκετές απ' αυτές τις έχουμε δει ξανά στην διδιάστατη περίπτωση και έτσι δεν θα αναφέρουμε περισσότερα. Τέτοιες είναι οι MeshStyle, Background, Epilog, ImageSize,PlotLabel,PlotRange,PlotRegion, DisplayFunction, AxesLabel, και Ticks. Θα αναφέρουμε την **View-**

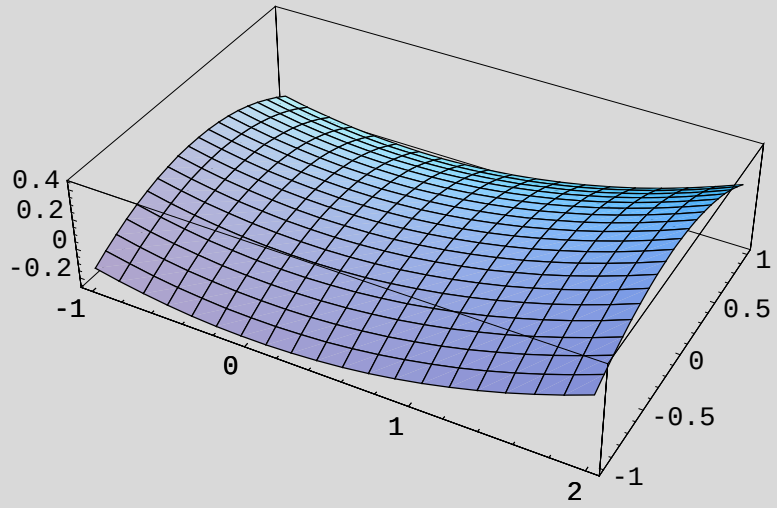
Point. Με την ViewPoint μπορούμε να δούμε με διαφορετική οπτική γωνία την ίδια επιφάνεια. Με τα πλήκτρα Ctrl Shift και V συγχρόνως πατημένα, παίρνουμε ένα κουτί με τους τρεις άξονες στο οποίο μπορούμε να επιλέξουμε το κατάλληλο σημείο(ViewPoint) του χώρου(απο το οποίο θα παρατηρούμε την επιφάνεια) και πατώντας Paste μπορούμε να το επικολήσουμε σε όποιο σημείο της Plot3D θέλουμε. Παρατηρείστε ότι η προεπιλεγμένη τιμή είναι ViewPoint->{1.3,-2.4,2.}. Με **Boxed->False** εξαφανίζουμε το διάφανο κουτί που περιβάλλει την επιφάνεια ενώ με **AxesStyle->...** μπορούμε να καθορίσουμε κάποιο "στύλ" με το οποίο θα σχδιαστούν οι τρείς κάθετοι άξονες. Παρακάτω σχεδιάζουμε το σαμάρι, εξαφανίζουμε το κουτί, θέτουμε Axes-Style→{GrayLevel[0.8], Thickness[0.01]}], θέτουμε επιγραφές σε κάθε άξονα και παρατηρούμε την επιφάνεια απο κάτω(ViewPoint->{-0.184, -1.819, -0.795}):

```
Plot3D@x^2ê9 - y^2ê4, 8x, -1, 2<, 8y, -1, 1<,
  Boxed → False, AxesStyle → 8GrayLevel@0.8D, Thickness@0.01D<,
  AxesLabel → 8"Ox", "Oy", "Oz"<,
  ViewPoint -> 8-0.184, -1.819, -0.795<D;
```



2. Τη γραφική παράσταση μιας επιφάνειας, που ορίζεται με την βοήθεια δύο παραμέτρων (π.χ u και v) την παίρνουμε με την **ParametricPlot3D**. Τέτοιες επιφάνειες λέγονται ***παραμετρικές*** επιφάνειες. Σε αυτήν την κατηγορία ανήκουν και επιφάνειες που ορίζονται με κυλινδρικές ή σφαιρικές συντεταγμένες. Αυτές οι επιφάνειες μπορεί να τέμνουν τον εαυτό τους ή να κουλουριάζουν γύρω απο κάποιο άξονα κ.ο.κ. Τέτοιες δυνατότητες δεν έχει η Plot3D. Μπορούμε να συμπεράνουμε λοιπόν εν κατακλείδι ότι η ParametricPlot3D είναι γενικότερη της Plot3D. Π.χ αν θέλουμε να σχεδιάσουμε το σαμάρι δεν έχουμε παρά να θέσουμε την x ίση με την πρώτη παράμετρο u και την y με την v και το z(δηλ. το ύψος f[x,y]) ίση με f[u,v]:

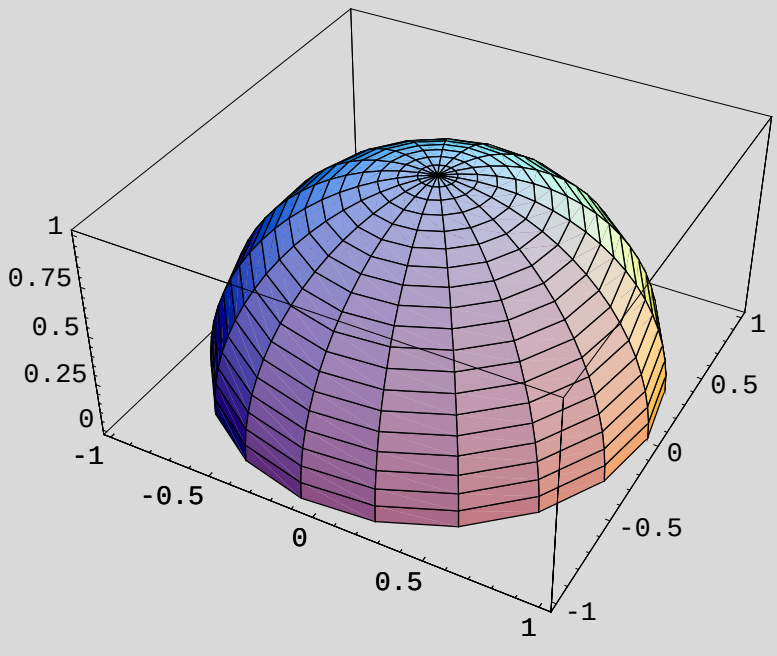
```
ParametricPlot3D@8u, v, u^2 ê 9 - v^2 ê 4<, 8u, -1, 2<, 8v, -1, 1<D;
```



Ως προς τις επιλογές διαφέρει ελάχιστα απο την Plot3D. Βασικά έχουμε διαφορετικές προεπιλεγμένες τιμές στα PlotPoints, και BoxRatios που είναι PlotPointsØAutomatic καιBoxRatiosØAutomatic.

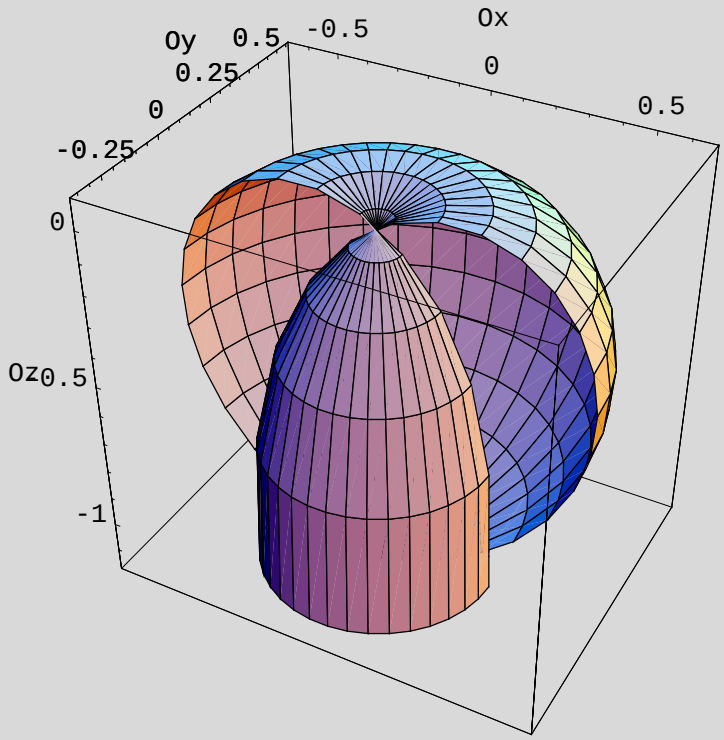
Με παραμετρικές εξισώσεις μπορούμε να ορίσουμε (και να σχεδιάσουμε κατά συνέπεια)τον τόρο, διάφορες κυλινδρικές επιφάνειες, παραβολοειδή, την σφαίρα κ.ο.κ. Παρακάτω δίνουμε την γραφική παράσταση ενός κομματιού της μοναδιαίας σφαίρας(με ακτινα ίση με 1) και με κέντρο την αρχή των αξόνων. Κατα'ρχην γράφουμε βέβαια την σφαίρα παραμετρικά με την χρήση των σφαιρικών συντεταγμένων!

```
ParametricPlot3D@
  8Cos@thetaD Sin@phiD, Sin@thetaD Sin@phiD, Cos@phiD<,
  8theta, 0 , 2 Pi<, 8phi, 0, Pi ê 2<D;
```

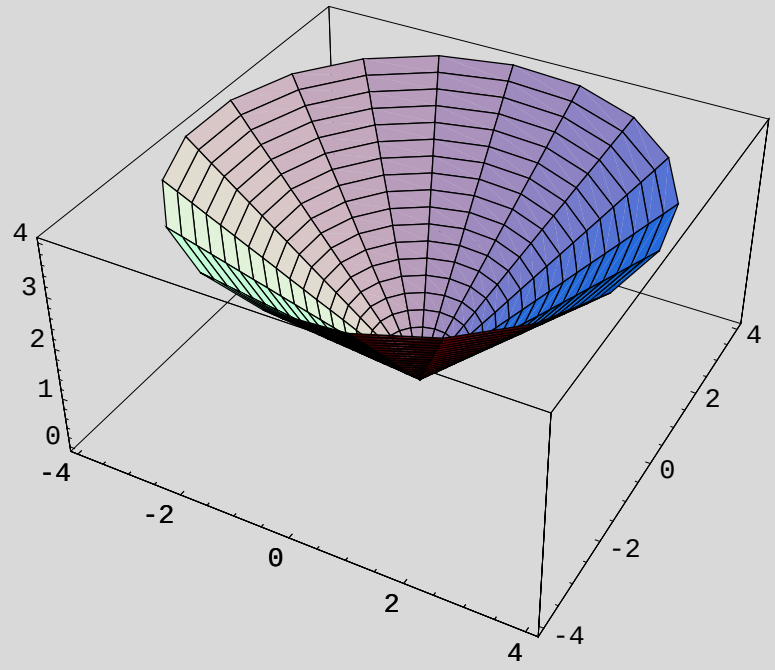


3. Ειδιαίτερα στις περιπτώσεις που η επιφάνεια έχει εξίσωση r=f[theta,phi](όπου r είναι η απόσταση του σημείου απο την αρχή των αξόνων) **σε σφαιρικές συντεταγμένες**, θα ήταν προτιμότερο να χρησιμοποιήσουμε την SphericalPlot3D. Όμοια, όταν η επιφάνεια μας ορίζεται με **κυλινδρικές συντεταγμένες** με την εξίσωση z=g[r,theta] όπου (r,theta,z οι κυλινδρικές συντεταγμένες,) τότε θα ήταν προτιμότερο να χρησιμοποιήσουμε την Cylindrical-Plot3D. Σε κάθε όμως περίπτωση πρέπει να καλέσουμε το πακέτο Graphics`ParametricPlot3D`. Π.χ

```
<< Graphics`ParametricPlot3D`
SphericalPlot3D@Log@thetaD, 8theta, Pi ê 10 , Pi<,
  8phi, 0, Pi<, AxesLabel → 8"Ox", "Oy", "Oz"<D;
```

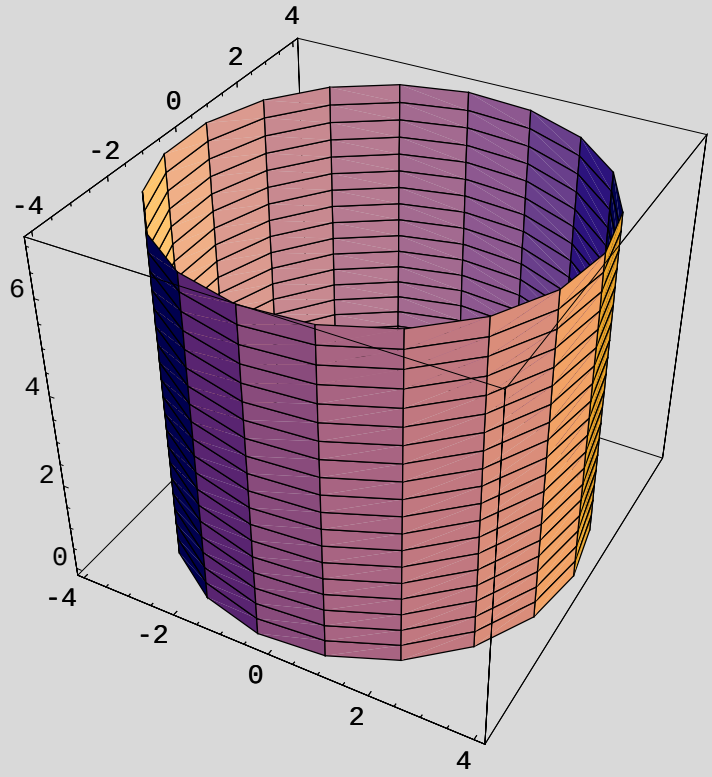


```
CylindricalPlot3D@r, 8r, 0, 4<, 8theta, 0, 2 Pi<D;
```



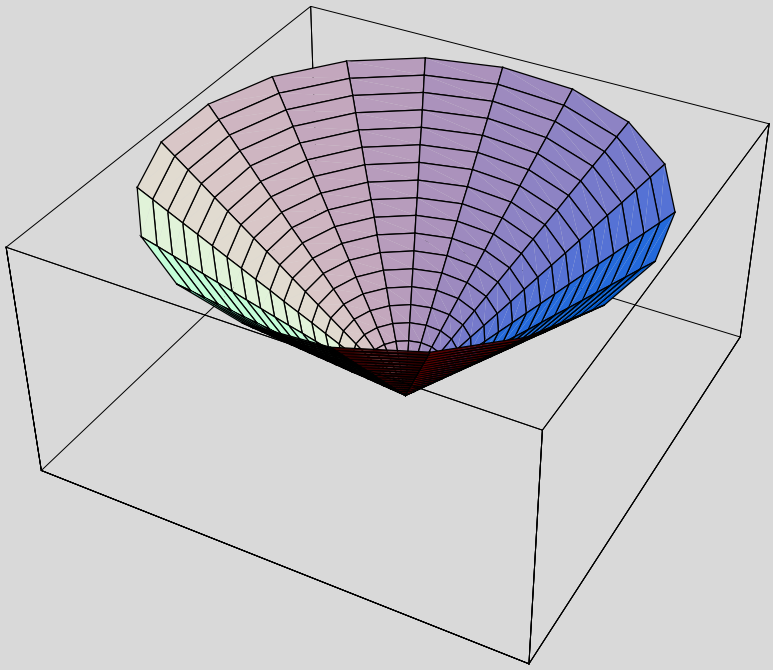
Παρατήρηση: Αν οι κυλινδρικές ή οι σφαιρικές επιφάνειες μας **δεν** δίνονται απο συναρτήσεις της μορφής που περιγράψαμε παραπάνω, τότε δεν μπορούμε να χρησιμοποιήσουμε τις CylindricalPlot3D και SphericalPlot3D ή θα πρέπει να λύσουμε ως προς z ή r αντίστοιχα! Για παράδειγμα, γνωρίζουμε ότι η εξίοσωση ενός κυλίνδρου σε κυλινδρικές συντεταγμένες δίνεται με την μορφή r= μια σταθερά και όχι με την μορφή συνάρτησης των r και theta. Σε τέτοιες περιπτώσεις θα πρέπει να χρησιμοποιήουμε την γνωστή μας ParametricPlot3D! Π.χ

```
ParametricPlot3D@
  84 Cos@thetaD, 4 Sin@thetaD, z<, 8theta, 0, 2 Pi<, 8z, 0, 7<D;
```



4. Βέβαια, ένας κύλινδρος είναι μια επιφάνεια η οποία έχει κάποιο άξονα, γύρω απο τον οποίο είναι τοποθετημένη συμμετρικά. Ορίζουμε τις **εκ περιστροφής επιφάνειες** εκείνες τις επιφάνειες που προκύπτουν απο την περιστροφή μιας επίπεδης καμπύλης γύρω απο μια ευθεία που λέγεται **άξονας**. Χωρίς περιορισμό της γενικότητας, μπορούμε να υποθέσουμε ότι ο άξονας περιστροφής είναι ο Oz και η καμπύλη βρίσκεται πάνω στο Oxz επίπεδο και δίνεται απο την εξίσωση z=f[x]. Τότε με την εντολή SurfaceOfRevolution[f[x],{x,xmin,xmax}] παίρνουμε την ζητούμενη επιφάνεια εκ περιστροφής.Για να εκτελεστεί η εντολή SurfaceOfRevolution θα πρέπει πρώτα να καλέσουμε το υποπρόγραμμα SurfaceOfRevolution του πακέτου Graphics. Π.χ για να σχεδιάσουμε ένα κώνο ύψους ίσο με 1 θα πρέπει να περιστρέψουμε την z=x δηλ. την διαγώνιο του Oxz επιπεδου γύρω απο τον Oz:

```
<< Graphics`SurfaceOfRevolution`
SurfaceOfRevolution@x, 8x, 0, 1<, Ticks → NoneD;
```

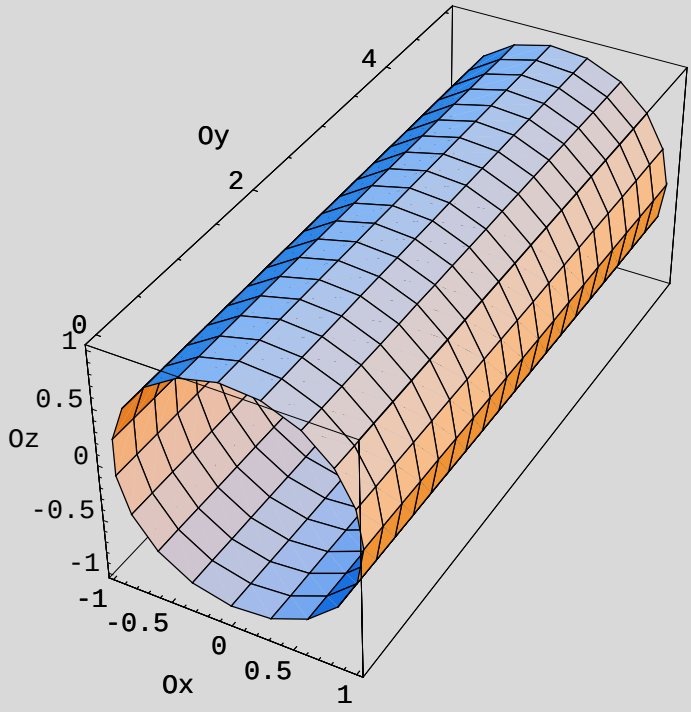


Για να σχεδιάσουμε ένα κύλινδρο γύρω απο τον Oz θα πρέπει να περιστρέψουμε την ευθεία x= κάποια σταθερά, γύρω απο τον Oz. Επειδή η καμπύλη με εξίσωση x=σταθερά **δεν** ειναι συνάρτηση του x θα χρειαστεί να χρησιμοποιήσουμε μια γενικοτερη μορφή της εντολής SurfaceOfRevolution. Ο γενικότερος τύπος είναι ο

SurfaceOfRevolution[{f[t],g[t],h[t]},{t,tmin,tmax},RevolutionAxis->{a,b,c}]

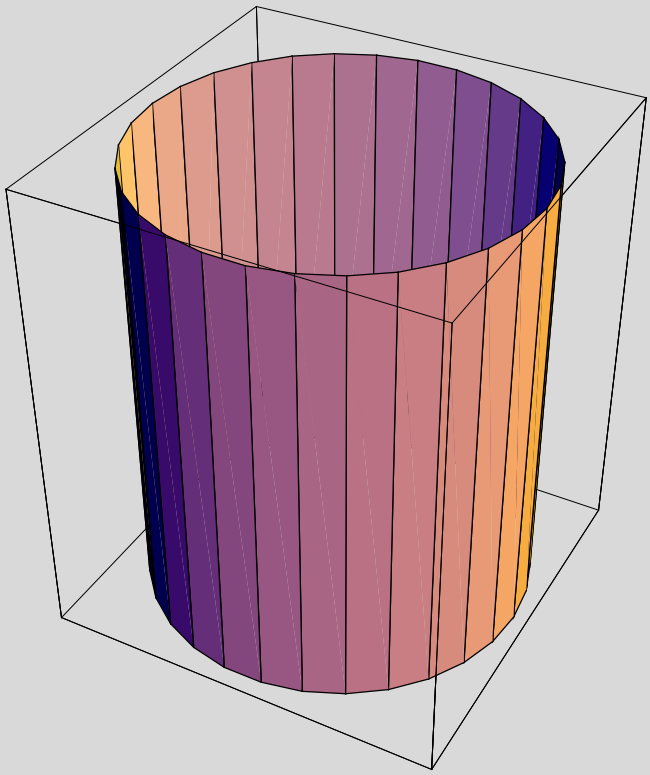
όπου τώρα η καμπύλη δίνεται παραμετρικά με τον τύπο {f[t],g[t],h[t]} όπου t είναι η παράμετρος. Επίσης με RevolutionAxis->{a,b,c} θέτουμε άξονα περιστροφής την ευθεία που διέρχεται απο την αρχή των αξόνων και το σημείο του χώρου με συντεταγμένες {a,b,c}. Π.χ για να σχεδιάσουμε τον κύλινδρο με ακτίνα ίση με 1 και μήκους 5 που περιστρέφεται γύρω απο τον άξονα Oy θα γράψουμε

```
SurfaceOfRevolution@81, t, 0<, 8t, 0, 5<,
  RevolutionAxis → 80, 1, 0<, AxesLabel → 8"Ox", "Oy", "Oz"<D;
```



5. Τελος θα αναφέρουμε ότι με την εντολή Show[Graphics3D[Σχήμα1[a,b,...],Σχήμα2[ ],...]] μπορούμε να πάρουμε άμμεσα και χωρίς πολύπλοκες πληκτρολογήσεις τη γραφική παράσταση κάποιων γνωστων Σχήμα(των). Οι σταθερές a,b,...είναι παράμετροι, οι τιμές των οποίων επηρεάζουν τη γραφική παράσταση. Για να δείτε ακριβώς πως, ζητείστε σχετική βοήθεια απο το Help. Το απαραίτητο πακέτο είναι το Graphics`Shapes. Το Σχήμα μπορεί να είναι έλικα, διπλή έλικα, κύλινδρος, κώνος, σφαίρα, λωρίδα του Mobius, πολύγωνα, ευθείες κ.ο.κ Π.χ για να σχεδιάσουμε ένα κύλινδρο ύψους 1.2 και ακτίνας 0.5 και για την σχεδίαση να χρησιμοποιηθούν 30 πολύγωνα θα γράψουμε:

```
<< Graphics`Shapes`
Show@Graphics3D@Cylinder@0.5, 0.6, 30DDD;
```



Κεφάλαιο 10ο: Διαδικασιακός Προγραμματισμός

10.1 Ανάθεση τιμών σε μεταβλητές

Ο τελεστής ανάθεσης (=, :=) χρησιμοποιείται για να τοποθετήσουμε το αποτέλεσμα μιας έκφρασης (τιμή ή παράσταση) σε μια μεταβλητή. Η σύνταξή του έχει ως εξής:

var = expr	Κατά τη διάρκεια εκτέλεσης της εντολής ο Η/Υ υπολογίζει πρώτα το αποτέλεσμα της έκφρασης expr και στη συνέχεια, αυτό που θα βρει, το αναθέτει στη μεταβλητή var που υπάρχει αριστερά του =. Στη συνέχεια του προγράμματος κάθε φορά που θα βλέπει την μεταβλητή var θα την αντικαθιστά από την τιμή έκφρασης expr.
Set[var, expr]:	Ισοδύναμη με την προηγούμενη ανάθεση τιμών.
var := expr	Η ανάθεση της τιμής της έκφρασης expr στην μεταβλητή var γίνεται την στιγμή που καλούμε την μεταβλητή var.
SetDelayed[var, expr]:	Ισοδύναμη με την προηγούμενη ανάθεση τιμών.

Το παραπάνω ίσον (=) λοιπόν και στις δύο περιπτώσεις αναφέρεται σε ανάθεση τιμής και όχι σε ισότητα. Το περιεχόμενο της μεταβλητής που υπήρχε πριν από την εντολή της ανάθεσης τιμής χάνεται.

```
A = Random@Integer, 81, 15<D;
B := Random@Integer, 81, 15<D;
A1 = Table@A, 85<D
B1 = Table@B, 85<D

85, 5, 5, 5, 5<
```

```
810, 11, 7, 6, 15<
```

Η πρώτη εντολή υπολογίζει έναν τυχαίο ακέραιο αριθμό μεταξύ 1 και 15 (π.χ. τον 7) και τον τοποθετεί στη μεταβλητή Α. Αντίθετα στη δεύτερη εντολή ορίζουμε ότι η μεταβλητή Β θα δεχτεί έναν τυχαίο ακέραιο αριθμό μεταξύ 1 και 15 όταν την καλέσουμε. Συνεπώς στην τρίτη εντολή επειδή η μεταβλητή Α έχει ήδη την τιμή 7, όταν προσπαθούμε να πάρουμε έναν πίνακα με 5 αριθμούς ίσους με Α, παίρνει και τους 5 ίσους με 7. Αντίθετα όταν προσπαθούμε να δημιουργήσουμε έναν πίνακα με 5 αριθμούς της μορφής Β, κάθε φορά που καλούμε τη μεταβλητή Β για να την τοποθετήσουμε στο πίνακα, υπολογίζεται η έκφραση που βρίσκεται δεξιά του ίσον στην δεύτερη εντολή και το αποτέλεσμα τοποθετείται στον πίνακα Β1. Αυτός είναι και ο λόγος που ο πίνακας Β1 έχει διαφορετικές τυχαίες τιμές.

```
Set@z, Expand@Hx + yL^2DD
```

$$x^2 + 2\,x\,y + y^2$$

```
SetDelayed@w, Expand@Hx + yL^2DD
w
```

$$x^2 + 2\,x\,y + y^2$$

Παρατηρούμε ότι το z και το w έχουν τις ίδιες τιμές. Αν όμως στη συνέχεια θέσουμε όπου x = 1 + a τότε θα έχουμε:

```
x = 1 + a
```

$$1 + a$$

```
z
```

$$H1 + aL^2 + 2\,H1 + aL\,y + y^2$$

```
w
```

$$1 + 2\,a + a^2 + 2\,y + 2\,a\,y + y^2$$

Η διαφορά αυτή οφείλεται στο γεγονός ότι το z έχει ήδη την τιμή του αναπτύγματος $Hx + yL^2$ δηλαδή $x^2 + 2\,xy + y^2$ και απλώς αντικαθιστά το x με ισοδύναμή του έκφραση 1 + a στο ανάπτυγμα αυτό. Αντίθετα όταν καλέσουμε το w τότε τοποθετείται στην έκφραση $Hx + yL^2$ το x = 1 + a και στη συνέχεια υπολογίζεται το ανάπτυγμα του $Hx + yL^2$.

```
A = 1
```

$$1$$

```
A = A + 1
```

$$2$$

Η πρώτη εντολή δημιουργεί μια θέση στην μνήμη του υπολογιστή για τη μεταβλητή Α και τοποθετεί σε αυτή την τιμή 1. Η δεύτερη εντολή βρίσκει το αποτέλεσμα δεξιά του ίσον (=) που είναι 1 + 1 (αφού η τιμή της μεταβλητής Α από την πρώτη εντολή είναι 1) και το τοποθετεί στην ίδια θέση μνήμης που έχει δημιουργήσει στην πρώτη εντολή για να αποθηκεύει την τιμή της μεταβλητής Α. Παρατηρούμε, δηλαδή, ότι οτιδήποτε βρίσκεται στην μεταβλητή που βρίσκεται αριστερά του ίσον, χάνεται και στη θέση του τοποθετείται αυτό που βρίσκεται δεξιά του ίσον.

Η εντολή A = A + 1 μπορεί να αντικατασταθεί με άλλες εντολές πιο σύντομες και μερικές φορές περισσότερο λειτουργικές:

a ++	Αυξάνει την τιμή της μεταβλητής a κατά 1, και επιστρέφει την προηγούμενη τιμή του a.
a --	Μειώνει την τιμή της μεταβλητής a κατά 1, και επιστρέφει την προηγούμενη τιμή του a.
++ a	Αυξάνει την τιμή της μεταβλητής a κατά 1, και επιστρέφει την νέα τιμή του a.
-- a	Μιεώνει την τιμή της μεταβλητής a κατά 1, και επιστρέφει την νέα τιμή του a.
a += da	Αυξάνει την τιμή της μεταβλητής a κατά da (δηλαδή a = a + da), και επιστρέφει την νέα τιμή του a.
a -= da	Μειώνει την τιμή της μεταβλητής a κατά da (δηλαδή a = a - da), και επιστρέφει την νέα τιμή του a.
a *= c	Πολλαπλασιάζει την τιμή της μεταβλητής a με c, τοποθετεί το αποτέλεσμα στην μεταβλητή a και επιστρέφει την νέα τιμή του a.
a /= c	Διαιρεί την τιμή της μεταβλητής a με c, τοποθετεί το αποτέλεσμα στην μεταβλητή a και επιστρέφει την νέα τιμή του a.

Ας δούμε μερικές εφαρμογές των παραπάνω εντολών:

```
A = 1;
A++
```

```
1
```

```
A
```

```
2
```

```
++A
```

```
3
```

```
A--
```

```
3
```

```
A
```

```
2
```

```
--A
```

```
1
```

```
A += 5
```

```
6
```

```
A -= 3
```

```
3
```

```
A  = 4
```

```
12
```

```
A ê= 6
```

```
2
```

Μπορούμε να αναθέσουμε την ίδια τιμή σε παραπάνω από μία μεταβλητές χρησιμοποιώντας την εντολή ανάθεσης:

var1 = var2 = ... = expr

```
x = y = 1
```

```
1
```

```
x + y
```

```
2
```

10.2 Εντολές Εισόδου - Εξόδου

Με τις εντολές εισόδου που διαθέτει το *Mathematica* μπορούμε να αναθέσουμε μία τιμή σε μια μεταβλητή πληκτρολογώντας την. Η συνάρτηση **Input** που διαθέτει το *Mathematica* μπορεί να χρησιμοποιηθεί ως μια εντολή εισόδου σε ένα πρόγραμμα.

var = **Input**[ ]	Ζητάει με διαλογική μορφή (μέσω παραθύρου) μια τιμή την οποία την αναθέτει στη μεταβλητή var.
var = **Input**["Μήνυμα"]	Ζητάει με διαλογική μορφή (μέσω παραθύρου) μια τιμή την οποία την αναθέτει στη μεταβλητή var, ενώ εμφανίζει οτ συγκεκριμένο μήνυμα για τον χρήστη στο παράθυρο διαλόγου.

Η εκτύπωση αποτελεσμάτων στο *Mathematica* γίνεται απλώς γράφοντας το όνομα της μεταβλητής που θέλουμε να εκτυπώσουμε είτε χρησιμοποιώντας την συνάρτηση **Print**[ ] που διαθέτει το *Mathematica.*

Print[έκφραση 1, έκφραση2 , ...]	Εκτυπώνει το σύνολο των εκφράσεων που υπάρχει μέσα στην Print και στη συνέχεια αλλάζει γραμμή.

Αν θέλουμε να εφαμόσουμε μια συγκεκριμένη μορφοποίηση στην εκτύπωσή μας μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **StylePrint**[ ].

Στη συνέχεια θα δώσουμε δύο απλά παραδείγματα εισόδου - εξόδου:

Παράδειγμα 1: Να γράψετε πρόγραμμα που να διαβάζει από το πληκτολόγιο δύο ακεραίους και να υπολογίζει και να εκτυπώνει το άθροισμα τους.

```
a = Input@"Give one Integer"D;
b = Input@"Give one Integer"D;
c = a + b;
Print@"Το άθροισμα είναι=", cD
```

Παράδειγμα 2: Να γράψετε πρόγραμμα που να διαβάζει από το πληκτολόγιο μία λίστα από πραγματικούς αριθμούς και να υπολογίζει και να εκτυπώνει τον μέσο όρο των στοιχείων της λίστας.

```
H a=Λίστα L
H mo=Μέσος όρος L
a = Input@"Δώσε τη Λίστα="D;
mo = Sum@a@@iDD, 8i, 1, Length@aD<D ê Length@aD;
Print@"Μέσος όρος=", moD
```

Στα παραπάνω παραδείγματα έχουμε να κάνουμε τις εξής παρατηρήσεις:
α. Οι εντολές δόθηκαν στο ίδιο cell (κελί) με την σειρά, πατώντας Enter στο τέλος κάθε γραμμής.
β. Τα σχόλια τοποθετούνται ανάμεσα στα σύμβολα (* και *).
γ. Το ερωτηματικό στο τέλος της εντολής σημαίνει ότι ναι μεν η εντολή εκτελείται αλλά το αποτέλεσμα δεν εμφανίζεται στην οθόνη.

10.3 Εντολές Ελέγχου

10.3.1 Η εντολή If

If [συνθήκη, εντολή(-ές) 1, εντολή(-ές)2, εντολή(-ές)3]

Αν η συνθήκη είναι αληθής, τότε θα εκτελεστεί η εντολή(-ές)1, αν είναι ψευδής θα εκτελεστεί η εντολή(-ές)2, ενώ αν δεν μπορεί το *Mathematica* να εκφέρει γνώμη για τον αν η συνθήκη είναι αληθής ή ψευδής εκτελείται η εντολή(-ές)3. Οι εντολές 2 και 3 είναι προεραιτικές.

Για την διατύπωση της συνθήκης στη σύνταξη της εντολής **If** θα χρειαστούμε τους παρακάτω τελεστές σύγκρισης και λογικούς τελεστές:

Τελεστές Σύγκρισης

Τελεστής	**Λειτουργία**
==	Ισότητα
!=	Ανισότητα
>	Μεγαλύτερο
<	Μικρότερο
>=	Μεγαλύτερο ή ίσο
<=	Μικρότερο ή ίσο

Όταν συντάσουμε τους παραπάνω τελεστές σύγκρισης στο *Mathematica*, το *Mathematica* αναλαμβάνει να τους ξαναγράψει με την εξής μορφή: **ã**, **ƒ**, >, <, **¥**, **§**.

Λογικοί Τελεστές

Τελεστής	**Λειτουργία**
&&	Σύζευξη (και)
||	Διάζευξη (ή)

Στη συνέχεια θα δώσουμε μερικά παραδείγματα εφαρμογής της εντολής **If**.

Παράδειγμα 3: Να κατασκευάσετε τη συνάρτηση:

$$f(x)=\begin{cases} x & x \neq 0 \\ 1 & x = 0 \end{cases}$$

```
f@x_D := If@x ≠ 0,
   1 ê x,
   0,
"άγνωστο"D;
```

```
f@2D
```

$$\frac{1}{2}$$

```
f@0D
```

```
0
```

```
f@qD
```

```
άγνωστο
```

Παρατηρούμε ότι αν η τιμή του ορίσματος x είναι διαφορετική του μηδενός (π.χ. 2) εκτελείται η πρώτη εντολή, ενώ αν η τιμή του ορίσματος x είναι μηδέν τότε εκτελείται η δεύτερη εντολή. Αν όμως θέσουμε στο x μια τιμή (π.χ. q) που το *Mathematica* δεν γνωρίζει αν αυτή είναι διάφορη ή ίση του μηδενός τότε εκτελείται η τρίτη εντολή.

Παράδειγμα 4: Έστω η εξίσωση πρώτου βαθμού ax + b = 0. Να γράψετε πρόγραμμα που θα υπολογίζει και θα εκτυπώνει τις πιθανές λύσεις της εξίσωσης.

```
a = Input@"Δώσε το a="D;
b = Input@"Δώσε το b="D;
If@a ≠ 0,
 x = -b ê a;
 Print@"x=", xD,
 If@b « 0,
  Print@"Αόριστη"D,
  Print@"Αδύνατη"DDD
```

Παράδειγμα 5: Να γράψετε πρόγραμμα που να διαβάζει από το πληκτολόγιο μία λίστα από ακεραίους αριθμούς και να υπολογίζει και να εκτυπώνει τη διάμεσο των αριθμών αυτών.

Για να βρούμε τη διάμεσο πρέπει πρώτα να ταξινομήσουμε τα στοιχεία της λίστας, έστω a, σε αύξουσα σειρά και στη συνέχεια αν το πλήθος των στοιχείων της λίστας, έστω n, είναι άρτιος τότε η διάμεσος είναι ίση με την τιμή (a(n/2)+a(n/2+1))/2 διαφορετικά είναι ίση με a((n+1)/2).

```
a = Input@"Δώσε τη Λίστα="D;
n = Length@aD;
s = Sort@aD;
If@EvenQ@nD,
  d = Hs@@n ê 2DD + s@@n ê 2 + 1DDL ê 2,
  d = s@@Hn + 1L ê 2DDD;
Print@"Διάμεσος=", dD
```

Παράδειγμα 6: Να γράψετε πρόγραμμα το οποίο να διαβάζει από το πληκτρολόγιο δύο ακεραίους αριθμούς και αν το τελευταίο τους ψηφίο είναι το ίδιο να εκτυπώνει το άθροισμα τους ενώ στην αντίθετη περίπτωση να εκτυπώνει την απόλυτη τιμή της διαφοράς τους.

```
a = Input@"Δώσε έναν ακέραιο"D;
b = Input@"Δώσε έναν ακέραιο"D;
mona = Mod@a, 10D;
monb = Mod@b, 10D;
If@mona == monb,
 Print@a + bD,
 If@a > b,
  Print@a - bD,
  Print@b - aDDD
```

Παράδειγμα 7: Να κατασκευάσετε συνάρτηση, η οποία θα δέχεται ως ορίσματα δύο ακεραίους αριθμούς και αν το τελευταίο τους ψηφίο είναι το ίδιο να εκτυπώνει το άθροισμα τους ενώ στην αντίθετη περίπτωση να εκτυπώνει την απόλυτη τιμή της διαφοράς τους.

```
f@a_, b_D := Hmona = Mod@a, 10D;
  monb = Mod@b, 10D;
  If@mona == monb,
   Print@a + bD,
   If@a > b,
    Print@a - bD,
    Print@b - aDDDL
```

```
f@24, 244D
```

```
268
```

```
f@23, 54D
```

```
31
```

```
f@54, 23D
```

```
31
```

Παράδειγμα 8: Έστω η εξίσωση δευτέρου βαθμού $ax^2 + bx + c = 0$. Να γράψετε πρόγραμμα που θα υπολογίζει και θα εκτυπώνει τις πιθανές λύσεις της εξίσωσης.

```
a = Input@"Δώσε το a="D;
b = Input@"Δώσε το b="D;
c = Input@"Δώσε το c="D;
If@a ≠ 0,
 d = b^2 - 4 a c;
 If@d > 0,
  x1 = H-b + Sqrt@dDL ê H2 aL;
  x2 = H-b - Sqrt@dDL ê H2 aL;
  Print@"x1=", x1, "x2=", x2D,
  If@d « 0,
   x = -b ê H2 aL;
   Print@"x=", xD,
   Print@"Μιγαδικές Λύσεις"DDD,
 If@b ≠ 0,
  x = -c ê b;
  Print@"x=", xD,
  If@c « 0,
   Print@"Αόριστη"D,
   Print@"Αδύνατη"DDDD
```

Παράδειγμα 9: Να κατασκευάσετε συνάρτηση, η οποία θα δέχεται ως ορίσματα τους συντελεστές (a, b, c) μιας δευτεροβάθμιας εξίσωση και εκτυπώνει τις λύσεις της εξίσωσης.

```
f@a_, b_, c_D := HIf@a ≠ 0,
   d = b^2 - 4 a c;
   If@d > 0,
    x1 = H-b + Sqrt@dDL ê H2 aL;
    x2 = H-b - Sqrt@dDL ê H2 aL;
    Print@"x1=", x1, "x2=", x2D,
    If@d « 0,
     x = -b ê H2 aL;
     Print@"x=", xD,
     Print@"Μιγαδικές Λύσεις"DDD,
   If@b ≠ 0,
    x = -c ê b;
    Print@"x=", xD,
    If@c « 0,
     Print@"Αόριστη"D,
     Print@"Αδύνατη"DDDDL
```

```
f@1, 1, 1D
```

```
Μιγαδικές Λύσεις
```

```
f@1, 2, 1D
```

```
x=-1
```

```
f@1, 5, 1D
```

$$x1=\frac{1}{2}\left(-5+\sqrt{21}\right)\quad x2=\frac{1}{2}\left(-5-\sqrt{21}\right)$$

10.3.2 Η εντολή Which

Which[συνθήκη1, εντολή(-ές) 1, συνθήκη2, εντολή(-ές)2, συνθήκη3, εντολή(-ές)3, ...]

Η εντολή **Which** ελέγχει τις παραπάνω συνθήκες "συνθήκη1", "συνθήκη2", "συνθήκη3" κ.λ.π. και στην πρώτη αληθή συνθήκη που θα βρεί, έστω τη συνθήκηi, θα εκτελέσει την αντίστοιχη εντολή(-ές)i.

Στη συνέχεια θα δώσουμε μερικά παραδείγματα εφαρμογής της εντολής **Which**.

Παράδειγμα 10: Να κατασκευάσετε συνάρτηση, η οποία θα δέχεται ως ορίσματα τις συντεταγμένες ενός σημείου και θα επιστρέφει τις τιμές 1, 2, 3 και 4 αν το σημείο βρίσκεται στο 1ο, 2ο, 3ο και 4ο τεταρτημόριο αντίστοιχα. Αν το σημείο είναι το (0, 0) να επιστρέφει την τιμή 0. Τέλος, αν το σημείο ανήκει στον άξονα των x να επιστρέφει την τιμή -1 ενώ αν το σημείο ανήκει στον άξονα των y να επιστρέφει την τιμή -2.

```
f@x_, y_D := Which@x « 0 && y « 0, 0,
   y « 0, -1,
   x « 0, -2,
   x > 0 && y > 0, 1,
   x < 0 && y > 0, 2,
   x < 0 && y < 0, 3,
   x > 0 && y < 0, 4D;
```

```
f@0, 0D
```

```
0
```

```
f@3, 0D
```

```
-1
```

```
f@0, 4D
```

```
-2
```

```
f@4, 5D
```

```
1
```

```
f@-2, 5D
```

```
2
```

```
f@-3, -2D
```

```
3
```

```
f@3, -2D
```

```
4
```

Παράδειγμα 11: Να γράψετε πρόγραμμα το οποίο να διαβάζει τον βαθμό πτυχίου ενός φοιτητή και να εκτυπώνει τον λεκτικό χαρακτηρισμό του.

```
a = Input@"Δωσε τον βαθμό πτυχίου"D;
Which@a ≤ 10 && a > 8.5, Print@"Άριστα"D,
 a ≤ 8.5 && a > 6.5, Print@"Πολύ καλά"D,
 a ≤ 6.5 && a ≥ 5, Print@"Καλώς"D,
 a > 10 »» a < 5, Print@"Λάθος ο βαθμός πτυχίου"DD
```

Παράδειγμα 12: Να κατασκευάσετε συνάρτηση, η οποία θα δέχεται ως όρισμα τον βαθμό πτυχίου ενός φοιτητή και να εκτυπώνει τον λεκτικό χαρακτηρισμό του.

```
f@a_D := Which@a ≤ 10 && a > 8.5, Print@"Άριστα"D,
  a ≤ 8.5 && a > 6.5, Print@"Πολύ καλά"D,
  a ≤ 6.5 && a ≥ 5, Print@"Καλώς"D,
  a > 10 »» a < 5, Print@"Λάθος ο βαθμός πτυχίου"DD
```

```
f@8.51D
```

```
Άριστα
```

```
f@6.51D
```

```
Πολύ καλά
```

```
f@6.5D
```

```
Καλώς
```

```
f@12D
```

```
Λάθος ο βαθμός πτυχίου
```

Παράδειγμα 13: Έστω η εξίσωση δευτέρου βαθμού $ax^2 + bx + c = 0$. Να γράψετε πρόγραμμα που θα υπολογίζει και θα εκτυπώνει τις πιθανές λύσεις της εξίσωσης με χρήση της εντολής **Which.**

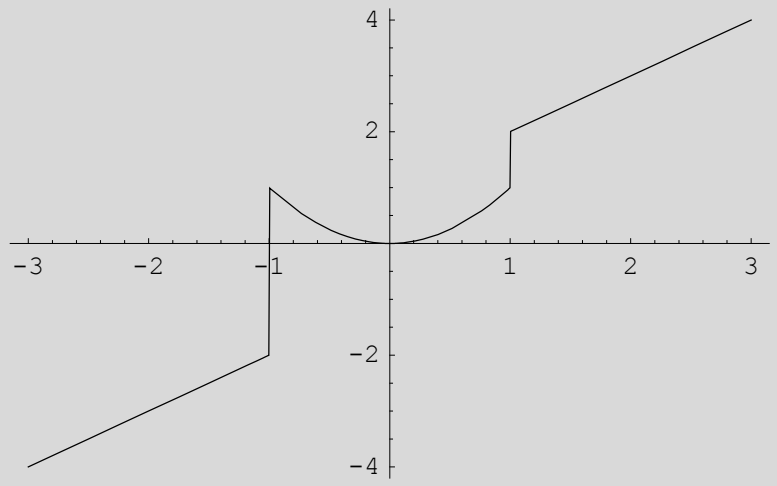
```
a = Input@"Δώσε το a="D;
b = Input@"Δώσε το b="D;
c = Input@"Δώσε το c="D;
If@a ≠ 0,
 d = b^2 - 4 a c;
 Which@d > 0,
  x1 = H-b + Sqrt@dDL ê H2 aL;
  x2 = H-b - Sqrt@dDL ê H2 aL;
  Print@"x1=", x1, "x2=", x2D,
  d « 0,
  x = -b ê H2 aL;
  Print@"x=", xD,
  d < 0,
  Print@"Μιγαδικές Λύσεις"DD,
 If@b ≠ 0,
  x = -c ê b;
  Print@"x=", xD,
  If@c « 0,
   Print@"Αόριστη"D,
   Print@"Αδύνατη"DDDD
```

Παράδειγμα 14: Να ορίσετε τη συνάρτηση

$$f(x) = \begin{cases} x - 1 & x \le 1 \\ x^2 - 1 & -1 < x < 1 \\ x + 1 & x \ge 1 \end{cases}$$

και να σχεδιάσετε τη γραφική της παράσταση για x∈[-3, 3].

```
f@x_D := Which@x ≤ -1, x - 1,
  x < 1, x^2,
  x ≥ 1, x + 1D
Plot@f@xD, 8x, -3, 3<D
```




```
- Graphics -
```

10.3.3 Η εντολή Switch

Κατά την εκτέλεση της εντολής **If** ο έλεγχος του προγράμματος μεταφέρεται σε ένα σύνολο εντολών ανάλογα με την τιμή μιας συνθήκης, η οποία μπορεί να πάρει δύο πιθανές τιμές: αληθής ή ψευδής.

Αν παρ' όλα αυτά θέλουμε να έχουμε παραπάνω από δύο επιλογές, τότε μπορούμε να χρησιμοποιήσουμε την εντολή Switch η οποία συντάσσεται ως εξής:

Switch[έκφραση, παράσταση1, εντολή(-ές) 1, παράσταση2, εντολή(-ές)2, παράσταση3, εντολή(-ές)3, ...]

Η εντολή **Switch** υπολογίζει αρχικά την έκφραση και στη συνέχεια ελέγχει με ποια από τις παραστάσεις "παράσταση1", "παράσταση2", παράσταση3" κ.λ.π. είναι ίση. Για την πρώτη ίση παράσταση που θα βρει, έστω τη παράστασηi θα εκτελέσει την αντίστοιχη εντολή(-ές).

Στη συνέχεια θα δώσουμε μερικά παραδείγματα εφαρμογής της εντολής **Switch**.

Παράδειγμα 15: Να κατασκευάσετε συνάρτηση, η οποία θα δέχεται ως όρισμα έναν ακέραιο αριθμό και θα ελέγχει αν ο αριθμός αυτός είναι άρτιος ή περιττός.

```
f@n_Integer?PositiveD := Switch@Mod@n, 2D,
  1, "Odd",
  0, "Even"D
```

```
f@2D
```

```
Even
```

```
f@1D

Odd
```

Η επικεφαλίδα Integer στο όρισμα της συνάρτησης αναγκάζει το n να είναι ακέραιος, ενώ ο έλεγχος ?Positive αναγκάζει το n να είναι και θετικός, αν θέλουμε η συνάρτηση f να δώσει απάντησει:

```
f@-10D

f@-10D
```

```
f@2.5D

f@2.5D
```

Παράδειγμα 16: Σε ένα παιχνίδι μεταξύ δύο ατόμων, η βαθμολογία έχει ως εξής:
α) Αν το άτομο Α παίξει την κίνηση 1 όταν το άτομο Β έπαιξε την κίνηση 1 τότε το άτομο Α κερδίζει 1 βαθμό.
β) Αν το άτομο Α παίξει την κίνηση 2 όταν το άτομο Β έπαιξε την κίνηση 1 τότε το άτομο Α χάνει 1 βαθμό.
γ) Αν το άτομο Α παίξει την κίνηση 1 όταν το άτομο Β έπαιξε την κίνηση 2 τότε το άτομο Α κερδίζει 2 βαθμούς.
δ) Αν το άτομο Α παίξει την κίνηση 2 όταν το άτομο Β έπαιξε την κίνηση 2 τότε το άτομο Α χάνει 2 βαθμούς.
Να κατασκευάσετε συνάρτηση που θα υπολογίζει το κέρδος του ατόμου Α ξέροντας τις κινήσεις των Α και Β.

```
f@x_, y_D := Switch@8x, y<,
  81, 1<, 1,
  82, 1<, -1,
  81, 2<, 2,
  82, 2<, -2D
```

Στην παραπάνω συνάρτηση αντιστοιχίσαμε τις κινήσεις x, y των παικτών Α και Β σε μία λίστα {x, y} και ανάλογα με τη τιμή της λίστα αυτής επιστρέφουμε και το αντίστοιχο κέρδος του παίκτη Α. Ας διαλέξουμε τώρα ένα τυχαίο παίξιμο για τους δύο παίκτες και στη συνέχεια ας υπολογίσουμε το κέρδος του παίκτη Α.

```
f@Random@Integer, 81, 2<D, Random@Integer, 81, 2<DD

2
```

10.4 Εντολές Επανάληψης

Σε πολλά προβλήματα απαιτείται η επανάληψη ενός συνόλου ενεργειών. Η επανάληψη μιας διαδικασίας παραπάνω από μία φορές ονομάζεται ανακύκλωση.

Στην ενότητα αυτή θα περιγράψουμε τις εντολές **Do**, **For** και **While** οι οποίες χρησιμοποιούνται σε προβλήματα που απαιτούνται ανακυκλώσεις. Συνήθως χρησιμοποιούμε τις εντολές **Do** και **For** όταν θέλουμε να επαναλάβουμε ένα σύνολο εντολών για συγκεκριμένο αριθμό επαναλήψεων ενώ την εντολή **While** την χρησιμοποιούμε όταν δεν είναι γνωστός εκ των προτέρων ο αριθμός των επαναλήψεων

10.4.1 Η εντολή Do

Do[εντολή, {imax}]	Εκτελεί την "εντολή" imax φορές.
Do[εντολή, {i, imax}]	Εκτελεί την "εντολή" καθώς η μεταβλητή i παίρνει τιμές από 1 έως imax με βήμα 1.
Do[εντολή, {i, imin, imax}]	Εκτελεί την "εντολή" καθώς η μεταβλητή i παίρνει τιμές από imin έως imax με βήμα 1.
Do[εντολή, {i, imin, imax, di}]	Εκτελεί την "εντολή" καθώς η μεταβλητή i παίρνει τιμές από imin έως imax με βήμα di.

Με τον όρο "εντολή" εννοούμε μία ή περισσότερες εντολές οι οποίες είναι χωρισμένες με ερωτηματικό.

Στην τελευταία από τις παραπάνω εκφράσεις της εντολής **Do** που είναι η πιο γενική μορφή διακρίνουμε τρεις περιπτώσεις, όσον αφορά το βήμα:

Περίπτωση 1η: di > 0.
Η μεταβλητή i παίρνει την i = imin.
Αν i **§** imax, τότε εκτελείται η "εντολή". Στη συνέχεια, προσθέτουμε το βήμα di στην τιμή της μεταβλητής i. Πηγαίνουμε πίσω και επαναλαμβάνουμε τον έλεγχο.
Αν i > imax, τότε μεταφερόμαστε εκτός της ανακύκλωσης.

Περίπτωση 2η: di < 0.
Η μεταβλητή i παίρνει την i = imin.
Αν i **¥** imax, τότε εκτελείται η "εντολή". Στη συνέχεια, προσθέτουμε το βήμα di στην τιμή της μεταβλητής i. Πηγαίνουμε πίσω και επαναλαμβάνουμε τον έλεγχο.
Αν i < imax, τότε μεταφερόμαστε εκτός της ανακύκλωσης.

Περίπτωση 3η: di = 0.
Δεν μπορεί το βήμα di να είναι μηδέν.

Ας δούμε στη συνέχεια μερικά παραδείγματα απλής εφαρμογής των παραπάνω εκφράσεων της εντολής **Do**:

```
Do@Print@"test"D, 85<D
```

```
test
test
test
test
test
```

```
Do@Print@iD, 8i, 1, 5<D
```

```
1
2
3
4
5
```

```
Do@a = i + 1; Print@aD, 8i, 5, 1, −1<D
```

```
6
5
4
3
2
```

Στη συνέχεια θα δώσουμε μερικά ακόμη παραδείγματα εφαρμογής της εντολής **Do**:

Παράδειγμα 17: Να δημιουργήσετε μία λίστα με 5 τυχαίους αριθμούς από το 1 έως το 20 και με 4 τυχαίους αριθμούς από το -20 έως το -1.

```
s = 8<;
Do@AppendTo@s, Random@Integer, 81, 20<DD, 85<D;
Do@AppendTo@s, Random@Integer, 8-20, -1<DD, 84<D;
s

810, 1, 3, 2, 14, -19, -1, -11, -13<
```

Παράδειγμα 18: Να κατασκευάσετε συνάρτηση, η οποία θα δημιουργεί μία λίστα με n τυχαίους αριθμούς από το 1 έως το 20 και με m τυχαίους αριθμούς από το -20 έως το -1.

```
tyxaioi@n_Integer?Positive, m_Integer?PositiveD := Hs = 8<;
  Do@AppendTo@s, Random@Integer, 81, 20<DD, 8n<D;
  Do@AppendTo@s, Random@Integer, 8-20, -1<DD, 8m<D;
  sL
```

```
tyxaioi@3, 4D

81, 19, 14, -18, -17, -6, -4<
```

Παράδειγμα 19: Να υπολογιστεί ο 10ος όρος της ακολουθίας

$$\alpha_n=\frac{1}{2}(\alpha_{n-1}+\frac{2}{\alpha_{n-1}}),\quad \alpha_1=1,$$

η οποία συγκλίνει στην τιμή $\sqrt{2}$.

```
aold = 1; DoAanew = 1/2 Haold + 2/aold L; aold = anew, 89<E; anew

4787633501779563550338751478164352626393810985192405254654229276259¥
  1925362787770306352384325384596398594331240032637710299217577668299¥
  6313024689222179880942725517434844559710363478381403509044255129799¥
 3385368114944226131160489088412764413184597197600430424080424489299¥
  1745564033552086544609184704239228339511303049317527075732707720¥
  4943610618917073241080260452775055121081948254768847591544963982¥
  848
```

Παρατηρούμε, ότι το αποτέλεσμα είναι ένας ρητός αριθμός. Πράγματι, αφού δώσαμε ως αρχική τιμή την ακέραια τιμή 1, το *Mathematica* κάνει πράξεις μεταξύ ακεραίων και προσπαθεί να δώσει αποτέλεσμα ακέραιο αριθμό. Επειδή όμως το πηλίκο δεν είναι ακέραιος αριθμός μας επιστρέφει ως αποτέλεσμα το ρητό αριθμό που αντιστοιχεί στο πηλίκο. Για να αποφύγουμε το παραπάνω αποτέλεσμα, μπορούμε είτε να χρησιμοποιήσουμε τη συνάρτηση **N** του *Mathematica* που επιστρέφει την αριθμητική τιμή της μεταβλητής anew είτε να δώσουμε ως αρχική τιμή τη πραγματική τιμή 1.0, οπότε το *Mathematica* κάνει πράξεις μεταξύ πραγματικών και δίνει αποτέλεσμα πραγματικό αριθμό.

```
aold = 1; DoAanew = 1/2 Jaold + 2/aold Ny; aold = anew, 89<E; N@anewD
```

```
1.41421
```

```
aold = 1.0; DoAanew = 1/2 Jaold + 2/aold Ny; aold = anew, 89<E; anew
```

```
1.41421
```

Παράδειγμα 20: Να κατασκευάσετε συνάρτηση, η οποία θα υπολογίζει τον n-στό όρο της ακολουθίας

$$\alpha_n=\frac{1}{2}(\alpha_{n-1}+\frac{2}{\alpha_{n-1}}),\quad \alpha_1=1,$$

η οποία συγκλίνει στην τιμή $\sqrt{2}$.

```
akolouthia@n_Integer?PositiveD :=
  Jaold = 1; DoAanew = 1/2 Jaold + 2/aold Ny; aold = anew, 8n - 1<E; N@anewDNy
```

```
akolouthia@10D
```

```
1.41421
```

Παράδειγμα 21: Να γράψετε πρόγραμμα που θα υπολογίζει τη σειρά: $S=1+2^2+3^3+4^4$.

```
s = 0;
Do@s = s + i^i, 8i, 1, 4<D;
s
```

```
288
```

Παράδειγμα 20: Να κατασκευάσετε συνάρτηση, η οποία θα υπολογίζει τη σειρα $S=1+2^2+3^3+4^4+...+n^n$ για δοθέντα ακέραιο n.

```
seira@n_Integer?PositiveD := Hs = 0;
  Do@s = s + i^i, 8i, 1, n<D;
  sL
```

```
seira@4D
```

```
288
```

Παράδειγμα 21: Να γράψετε πρόγραμμα που θα διαβάζει έναν μονοδιάστατο πίνακα ακεραίων αριθμών και θα υπολογίζει και θα εκτυπώνει το πλήθος των θετικών και των αρνητικών αριθμών του πίνακα.

```
a = Input@"Δώσε τον πίνακα"D;
negcounter = poscounter = 0;
Do@If@a@@iDD ≥ 0,
   poscounter = poscounter + 1,
   negcounter = negcounter + 1D,
  8i, 1, Length@aD<D;
Print@"poscounter=", poscounterD;
Print@"negcounter=", negcounterD;
```

10.4.1.1 Εμφωλευμένες επαναληπτικές διαδικασίες (ανακυκλώσεις)

Είναι δυνατό να τοποθετήσουμε μια **Do** μέσα σε μία άλλη, ή αλλιώς να έχουμε δύο μεταβλητές αντί για μία μέσα στην **Do** όπως φαίνεται παρακάτω:

Do[εντολή, {i, imin, imax, di}, {j, jmin, jmax, dj}] ή **Do**[**Do**[εντολή, {j, jmin, jmax, dj}], {i, imin, imax, di}]

Αρχικά θέτει i=imin και εκτελεί την "εντολή" καθώς η μεταβλητή j παίρνει τιμές από jmin έως jmax με βήμα dj. Εφόσον τελειώσουν οι πρώτες επαναλήψεις, προσθέτει στη μεταβλητή i το βήμα di και επαναλαμβάνει την εκτέλεση της "εντολής" καθώς η μεταβλητή j παίρνει τιμές από jmin έως jmax με βήμα dj. Η διαδικασία συνεχίζεται έως ότου η τιμή της μεταβλητής i ξεπεράσει το imax.

Με τα παρακάτω παραδείγματα, μπορούμε εύκολα να καταλάβουμε πως λειτουργεί η εντολή **Do** όταν έχουμε διπλή επανάληψη (ανακύκλωση):

```
Do@Print@8i, j<D, 8i, 1, 3<, 8j, 1, 2<D
```

```
81, 1<
```

```
81, 2<
```

```
82, 1<
```

```
82, 2<
```

```
83, 1<
```

```
83, 2<
```

```
Do@
 Do@Print@8i, j<D, 8j, 1, 2<D,
 8i, 1, 3<D
```

```
81, 1<
```

```
81, 2<
```

```
82, 1<
```

```
82, 2<
```

```
83, 1<
```

```
83, 2<
```

Στα παραπάνω δύο παραδείγματα, για i=1 εκτελείται η εντολή Print[{i,j}] για j=1, 2, 3. Στη συνέχεια αυξάνεται το i κατά ένα, i=2 και εκτελείται η "εντολή" Print[{i,j}] για j=1, 2, 3.

Παράδειγμα 22: Να γράψετε πρόγραμμα που θα διαβάζει τα στοιχεία ενός δισδιάστατου πίνακα ακεραίων αριθμών και:
i) θα διαβάζει έναν αριθμό που θα αντιστοιχεί σε μια στήλη και θα υπολογίζει το ελάχιστο της στήλης αυτής και
ii) θα διαβάζει έναν αριθμό που θα αντιστοιχεί σε μια γραμμή και θα υπολογίζει το μέγιστο της γραμμής αυτής.

```
a = Input@"Δώσε πίνακα="D;
r = Input@"Δώσε τη γραμμή="D;
c = Input@"Δώσε τη στήλη="D;
max = a@@r, 1DD;
Do@If@max < a@@r, jDD,
   max = a@@r, jDDD,
  8j, 1, Length@a@@rDDD<D;
Print@"Το max της ", r, " γραμμής είναι:", maxD;
min = a@@1, cDD;
Do@If@min > a@@i, cDD,
   min = a@@i, cDDD,
  8i, 1, Length@aD<D;
Print@"Το min της ", c, " στήλης είναι:", minD;
```

Παράδειγμα 23: Να γράψετε πρόγραμμα που θα διαβάζει τα στοιχεία ενός δισδιάστατου πίνακα ακεραίων αριθμών και θα εξετάζει αν ο πίνακας είναι αραιός. Θεωρούμε ότι ένας πίνακας είναι αραιός αν πάνω από το 80% του πλήθους των στοιχείων του είναι μηδέν.

```
a = Input@"Δώσε πίνακα="D;
counter = 0;
Do@
  Do@If@a@@i, jDD « 0,
    counter = counter + 1D,
   8j, 1, Length@a@@1DDD<D,
  8i, 1, Length@aD<D;
If@counter > 0.8  Length@aD  Length@a@@1DDD,
  Print@"Αραιός"D,
  Print@"Όχι αραιός"DD;
```

10.4.2 Η εντολή For

Η σύνταξη της εντολής **For** μοιάζει πολύ με τη σύνταξη της εντολής **For** που συναντάμε στην γλώσσα προγραμματισμού C με τη μόνη διαφορά ότι αντί για το σύμβολο ";" χρησιμοποιούμε το σύμβολο ",".

For[έναρξη, έλεγχος, αύξηση, εντολή]

Το πρώτο όρισμα "έναρξη" είναι αυτό που θα εκτελεστεί πρίν αρχίσει η επάναληψη. Το δεύτερο όρισμα "έλεγχος" περιέχει έναν έλεγχο ο οποίος θα γίνεται πριν την κάθε επανάληψη. Αν ο έλεγχος επιστρέψει True τότε θα ακολουθήσει η επανάληψη διαφορετικά η επανάληψη θα σταματήσει. Το τρίτο όρισμα "αύξηση" εκτελείται μετά την εντολή και έχει σκοπό την ενημέρωση κάποιας μεταβλητής που περιέχεται στο όρισμα "έλεγχος". Συνήθως πρόκειται για προσαύξηση του μετρητή επανάληψης κατά ένα βήμα. Το τέταρτο όρισμα "εντολή" αποτελείται από μία ή περισσότερες εντολές οι οποίες θα εκτελεστούν κατά την επανάληψη. Οι εντολές αυτές θα πρέπει να είναι χωρισμένες με ερωτηματικό.

```
For@i = 1, i ≤ 5, i++, Print@iDD
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

Παράδειγμα 24: Να δημιουργήσετε μία λίστα με 5 τυχαίους αριθμούς από το 1 έως το 20 και με 4 τυχαίους αριθμούς από το -20 έως το -1. (Να κάνετε χρήση της εντολής For).

```
s = 8<;
For@i = 1, i ≤ 5, i++, AppendTo@s, Random@Integer, 81, 20<DDD;
For@i = 1, i ≤ 4, i++, AppendTo@s, Random@Integer, 8−20, −1<DDD;
s

811, 16, 3, 7, 10, −5, −1, −5, −8<
```

Παράδειγμα 25: Μια πολύ γνωστή ακολουθία φυσικών αριθμών στα μαθηματικά είναι η ακολουθία Fibonacci (ορισμένη από τον Ιταλό μαθηματικό Leonardo Fibonacci). Οι όροι της ακολουθίας αυτής, ας τη συμβολήσουμε f, ορίζονται ως εξής:

$$f_1 = 1,\ f_2 = 1,\ f_3 = 2,\ f_4 = 3,\ f_5 = 5 \ldots$$

και γενικώς $f_1 = f_{n-1} + f_{n-2}$ (δηλαδή κάθε όρος εκτός από τους δύο πρώτους, δίνεται ως άθροισμα των δύο προηγούμενων όρων). Γράψτε πρόγραμμα που θα ζητά ένα φυσικό αριθμό n και στη συνέχεια θα υπολογίζει και θα εκτυπώνει τον n-στό όρο της ακολουθίας Fibonacci. (Να κάνετε χρήση της εντολής For).

```
n = Input@"Δώσε έναν φυσικό αριθμό"D;
f1 = 1;
f2 = 1;
For@i = 3, i ≤ n, i++,
  f = f2 + f1;
  f1 = f2;
  f2 = fD;
Print@"O ", n, "−στός όρος της ακολουθίας Fibonnaci είναι ο:"D
Print@fD
```

```
O 150−στός όρος της ακολουθίας Fibonnaci είναι ο:
```

```
9969216677189303386214405760200
```

Παράδειγμα 26: Να κατασκευάσετε συνάρτηση, η οποία θα υπολογίζει τον n-οστό όρο της ακολουθίας fibonacci. (Να κάνετε χρήση της εντολής For).

```
fibo@n_Integer?PositiveD := Hf1 = 1;
  f2 = 1;
   For@i = 3, i ≤ n, i++,
    f = f2 + f1;
    f1 = f2;
    f2 = fD;
  fL
```

```
fibo@10D

55
```

Όπως και στην εντολή **Do** έτσι και στην εντολή **For** (όπως και σε κάθε εντολή επανάληψης) μπορούμε να τοποθετήσουμε μια **For** μέσα σε μία άλλη:

```
For@i = 1,
 i ≤ 2,
 i++,
 For@j = 1,
  j ≤ 3,
  j++,
  Print@8i, j<DDD
```

```
81, 1<
```

```
81, 2<
```

```
81, 3<
```

```
82, 1<
```

```
82, 2<
```

```
82, 3<
```

Παράδειγμα 27: Να γράψετε πρόγραμμα που θα διαβάζει τα στοιχεία ενός δισδιάστατου πίνακα ακεραίων αριθμών και θα εξετάζει αν ο πίνακας είναι αραιός. Θεωρούμε ότι ένας πίνακας είναι αραιός αν πάνω από το 80% του πλήθους των στοιχείων του είναι μηδέν. (Να κάνετε χρήση της εντολής For).

```
a = Input@"Δώσε πίνακα="D;
counter = 0;
For@i = 1, i ≤ Length@aD, i++,
  For@j = 1, j ≤ Length@a@@1DDD, j++, If@a@@i, jDD « 0,
    counter = counter + 1DDD;
If@counter > 0.8  Length@aD  Length@a@@1DDD,
  Print@"Αραιός"D,
  Print@"Οχι αραιός"DD;
```

Παράδειγμα 28: Να υπολογίσετε το άθροισμα των αρτίων αριθμών από το 1 έως το 1000. (Να κάνετε χρήση της εντολής For).

```
s = 0;
For@i = 2, i ≤ 1000, i = i + 2, s = s + iD;
s

250500
```

Παράδειγμα 29: Να υπολογίσετε το άθροισμα των περιττών αριθμών από το 1 έως το 1000. (Να κάνετε χρήση της εντολής For).

```
s = 0;
For@i = 1, i ≤ 1000, i = i + 2, s = s + iD;
s

250000
```

10.4.3 Η εντολή While

Η εντολή **While** χρησιμοποιείται συνήθως όταν θέλουμε να επαναλάβουμε ένα σύνολο εντολών για έναν αριθμό φορών που δεν γνωρίζουμε εκτ ων προτέρων.

While[έλεγχος, εντολή]

Πρώτα γίνεται η εκτέλεση του "ελέγχου". Αν επιστραφεί False η επαναληπτική διαδικασία σταματάει. Αν επιστραφεί True τότε εκτελείται η "εντολή" και επαναλαμβάνεται ο "έλεγχος". Το δεύτερο όρισμα "εντολή" αποτελείται από μία ή περισσότερες εντολές οι οποίες θα εκτελεστούν κατά την επανάληψη. Οι εντολές αυτές θα πρέπει να είναι χωρισμένες με ερωτηματικό.

Θα πρέπει να δωθεί μεγάλη προσοχή ώστε οι μεταβλητές που περιέχονται στον "έλεγχο" να έχουν πάρει αρχική τιμή πριν τον "έλεγχο" ώστε να μπορέσει να εκτελεστεί η "εντολή", αλλά και να αλλάζουν τιμή μέσα στην "εντολή" ώστε η επαναληπτική διαδικασία να σταματήσει σε κάποιο πεπερασμένο βήμα.

Παράδειγμα 30: Να υπολογίσετε το άθροισμα των αρτίων αριθμών από το 1 έως το 1000. (Να κάνετε χρήση της εντολής While).

```
s = 0;
i = 2;
While@i ≤ 1000, s = s + i; i = i + 2D
s

250500
```

Παράδειγμ 31: Να υπολογίσετε τη ρίζα της συνάρτησης $f[x] = \cos[x] - x^2$ με ακρίβεια $e=10^{-6}$κάνοντας χρήση της επαναληπτικής μεθόδου Newton-Rapshon:

$$\alpha_{n+1}=\alpha_n-\frac{f[\alpha_n]}{f'[\alpha_n]},\quad \alpha_0 = c \in R.$$

```
f@x_D = Cos@xD - x^2;
x0 = Input@"Δώσε αρχική τιμή="D;
If@f'@x0D « 0,
  Print@"Error"D,
  xold = N@x0D; xnew = N@xold - f@xoldD ê f'@xoldDD;
  While@Abs@xnew - xoldD > 10^-6,
   xold = xnew; xnew = N@xold - f@xoldD ê f'@xoldDDDD;
xnew

0.824132
```

Παράδειγμ 32: Να γράψετε πρόγραμμα το οποίο θα υπολογίζει τη σειρά S=$\frac{1}{2}+\frac{1}{4}+\frac{1}{6}+\frac{1}{8}$+... με τελευταίο όρο αυτόν που δεν ξεπερνάει την τιμή 0.000001.

```
s = 0;
i = 1;
While@H1. ê H2  iL > 0.00001L, s = s + 1. ê H2  iL; i = i + 1D;
s

5.69849
```

Παράδειγμ 33: Να γράψετε πρόγραμμα το οποίο θα υπολογίζει τον μέγιστο κοινό διαιρέτη δύο αριθμών σύμφωνα με τον αλγόριθμό του Ευκλείδη.

```
a = Input@"Δώσε έναν ακέραιο"D;
b = Input@"Δώσε έναν ακέραιο"D;
If@a ≥ b, diaireteos = a;
  diairetis = b, diaireteos = b; diairetis = aD;
r = Mod@diaireteos, diairetisD;
While@r ≠ 0, diaireteos = diairetis;
  diairetis = r; r = Mod@diaireteos, diairetisDD;
diairetis
```

Παράδειγμ 34: Να κατασκευάσετε συνάρτηση η οποία θα υπολογίζει τον μέγιστο κοινό διαιρέτη δύο αριθμών σύμφωνα με τον αλγόριθμό του Ευκλείδη.

```
mkd@a_Integer, b_IntegerD :=
 HIf@a ≥ b, diaireteos = a;
   diairetis = b, diaireteos = b; diairetis = aD;
  r = Mod@diaireteos, diairetisD;
  While@r ≠ 0, diaireteos = diairetis;
   diairetis = r; r = Mod@diaireteos, diairetisDD;
  diairetisL
```

```
mkd@320, 180D

20
```

Για επαλήθευση μπορούμε να χρησιμοποιήσουμε και τη συνάρτηση GCD που διαθέτει το *Mathematica*.

```
GCD@320, 180D

20
```