

Γράψτε μια συνάρτηση που να έχει ως παράμετρο έναν μερικώς συμπληρωμένο πίνακα χαρακτήρων και η οποία να διαγράφει όλα τα επαναλαμβανόμενα γράμματα του πίνακα. Όταν ένα γράμμα διαγράφεται, τότε τα υπόλοιπα γράμματα μετακινούνται μπροστά για να καλύψουν το κενό. Επειδή ένας μερικώς συμπληρωμένος πίνακας απαιτεί δύο ορίσματα η συνάρτηση θα έχει δύο παραμέτρους: μια για τον πίνακα και μια τύπου int για τον αριθμό των θέσεων που χρησιμοποιούνται. Επίσης η δεύτερη παράμετρος θα είναι παράμετρος με αναφορά καθώς θα αλλάζει κατά την διάρκεια των αλλαγών.

```
void change (char A[], int &size)
```

```
int main ( )
```

```
{
```

```
const int BIG=10;
```

```
char A[BIG], next;
```

```
int i=0, size;
```

```
cin >> next;
```

```
while ( (i < BIG) && (next != ".") )
```

```
{
```

```
    A[i] = next;
```

```
    cin >> next;
```

```
    i++;
```

```
}
```

```
size = i;
```

```
change (A, size);
```

```
for (i=0; i < size; i++)
```

```
    cout << A[i];
```

```
void change (char A[], int &size)
```

```
{ int i, j, k;
```

```
  for (i=0; i < size-1; i++)
```

```
  { for (j=i+1; j < size; j++)
```

```
    { if (A[j] == A[i];
```

```
      { for (k=j; k < size; k++)
```

```
        { A[k] = A[k+1];
```

```
          size = size-1
```

```
          j = j-1; } } } }
```

- 3) Να συμπληρώσετε τα 20 στοιχεία ενός μονοδιάστατου πίνακα με τυχαίες τιμές (rand()) και στη συνέχεια να βρείτε
- το άθροισμα των τυχαίων στοιχείων
  - το μέσο όρο τους
  - το μέγιστο & το ελάχιστο στοιχείο του πίνακα.

ΛΥΣΗ

```
float mo(int a[], int size)
int sum(int a[], int size)
int max(int a[], int size)
int min(int a[], int size)
int main()
```

```
{
    int a[20], i;
    for (i=0; i<20; i++)
        { a[i]=rand(); }
    for (i=0; i<20; i++)
        { cout << a[i]; }
    cout << max(a, 20);
    cout << min(a, 20);
    cout << mo(a, 20);
    cout << sum(a, 20);
    return 0;
}
```

```
int sum(int a[], int size)
{
    int sum=0; i;
    for (i=0; i<20; i++)
        { sum=sum+a[i]; }
    return sum;
}
```

```
int max(int a[], int size)
{
    int max, i;
    max=a[0];
    for (i=0; i<20; i++)
        { if (a[i]>max)
            return max;
        }
}
```

```
int min(int a[], int size)
{
    int min, i;
    min=a[0];
    for (i=0; i<20; i++)
        { if (a[i]<min)
            return min;
        }
}
```

```
float mo(int a[], int size)
{
    return (float(sum(a, 20)/size));
}
```

2) Ένας αριθμός λέγεται παλίνδρομος αν μπορεί να διαβαστεί το ίδιο από την αρχή και το τέλος του (πχ 2772 ή 4444 κ.λπ).  
 Να γίνει συνάρτηση που να δέχεται έναν ακέραιο τετραψήφιο αριθμό και επιστρέφει true ή false αναλόγως αν ο αριθμός είναι παλίνδρομος ή όχι. Στη main ( ) θα διαβαστεί ο αριθμός θα πρέπει να ελεγχτείτε κατά την είσοδο του αριθμού επαναληπτικά αν αυτός είναι 4ψήφιος.

ΛΥΣΗ

```
#include <iostream>
using namespace std;
bool Palindromos (int a)
int main ( )
{
    int a, Pal;
    cin >> a;
    do
        { cin >> a; }
    while ((a < 1000) || (a > 9999));
    Pal = Palindromos (a);
    cout << Palindromos (a);
    return 0;
}
```

```
bool Palindromos (int a)
{
    int arithmos, Pal1, psifia;
    bool flag;
    arithmos = a;
    Pal1 = 0;
    while (a != 0)
    {
        psifia = a % 10;
        a = a / 10;
        Pal1 = 10 * Pal1 + psifia;
    }
    if (arithmos == Pal1)
        flag = true;
    else
        flag = false;
    return (flag);
}
```

1) Να γίνει πρόγραμμα όπου να βρεθεί το μέσο όρο βαθμών από ένα διαγώνισμα σε μια τάξη (n) μαθητών, με ακέραιο βαθμό από το 0 ως το 100.

i) Για γνωστό πλήθος μαθητών (n=10)

ii) Για άγνωστο πλήθος μαθητών

ΛΥΣΗ

```
i) int main ( )
{
    int Grade, sumGrade=0;
    double avg;
    for (int i=1; i<=10; i++)
    { do
        { cin >> Grade; }
      while ( (Grade < 0) || (Grade > 100) );
      sumGrade = sumGrade + Grade;
    }
    avg = sumGrade / 10.0;
    cout << avg;
    return 0;
}
```

```
ii) int main ( )
{
    int Grade, sumGrade=0, I=0, ENDNumber;
    double avg;
    cin >> Grade > ENDNumber;
    while (Grade != ENDNumber)
    { I = I + 1;
      sumGrade = sumGrade + Grade;
      cin >> Grade;
    }
    if ( I != 0 )
    { avg = sumGrade / I; }
    else
    cout << "FALSE";
    return 0;
}
```