Άσκηση 1

- Να γραφεί ένα πρόγραμμα το οποίο να αρχικοποιεί έναν μονοδιάστατο πίνακα με τους αριθμούς 5,8,9,4,2,1,7,5,6,2 και κατόπιν να μετατρέπει τους μονούς σε ζυγούς. Τέλος να εμφανίζει τα περιεχόμενα του πίνακα.

```
#include <iostream>
using namespace std;
int main()
{
        int a[10]={5,8,9,4,2,1,7,5,6,2},i;
        for (i=0;i<10;i++)
                if (a[i]%2!=0) a[i]++;
        for (i=0;i<10;i++)
                cout << a[i] << endl;
   return 0;
}
```

Άσκηση 2

- Να γραφεί ξανά ο κώδικας του προηγούμενου προγράμματος με χρήση δεικτών για την προσπέλαση των θέσεων μνήμης του πίνακα

```
#include <iostream>
using namespace std;
int main()
{
        int a[10]={5,8,9,4,2,1,7,5,6,2},i;
        for (i=0;i<10;i++)
                if (*(a+i)%2!=0) (*(a+i))++;
        for (i=0;i<10;i++)
                cout << *(a+i) << endl;
    return 0;
}
```

Άσκηση 3

- Να γραφεί ξανά το πρόγραμμα της προηγούμενης άσκησης ώστε η πρόσθεση της μονάδας στους ζυγούς του πίνακα να γίνεται από μια συνάρτηση στην οποία θα μεταβιβάζεται ο πίνακας των αριθμών

```
#include <iostream>
using namespace std;
void do_it(int p[]);
int main()
{
	int a[10]={5,8,9,4,2,1,7,5,6,2},i;
	do_it(a);
	for (i=0;i<10;i++)
		cout << a[i] << endl;
	return 0;
}
void do_it(int p[])
{
	int i;
	for (i=0;i<10;i++)
		if (p[i]%2!=0) p[i]++;
}
```

Άσκηση 4

- Να γραφεί πρόγραμμα το οποίο να συμπληρώνει έναν πίνακα 20 θέσεων με τυχαίους ακέραιους αριθμούς (συνάρτηση rand()). Κατόπιν, το πρόγραμμα πρέπει να βρίσκει τον μεγαλύτερο αριθμό του πίνακα, καθώς και τη θέση στην οποία βρίσκεται μέσα στον πίνακα

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
	int a[20],i,max,thesi;
	for (i=0;i<20;i++)
		a[i]=rand();
	max=a[0];
	thesi=1;
	for (i=0;i<20;i++)
	{
		if (a[i]>max)
		{
			max=a[i];
			thesi=i;
		}
	}
	cout << "Μέγιστος "<<max <<" στη θέση "<<thesi<<endl;
	return 0;
}
```

Άσκηση 5

- Με δεδομένο έναν πίνακα int a[100][20] να γραφεί κώδικας ο οποίος να εμφανίζει τον μεγαλύτερο αριθμό από κάθε γραμμή του πίνακα

```
#include <iostream>
using namespace std;
int main()
{
   int a[100][20],max;
   int i,j;
   .....
   for (i=0;i<100;i++)
   {
      max=a[i][0];
      for (j=0;j<20;j++)
      {
         if (a[i][j]>max)
            max=a[i][j];


       }
       cout << "Μέγιστος σειράς " << i << "=" << max;
    }
    return 0;
}
```

Η αρχική τιμή της **max** είναι κάθε φορά η τιμή της πρώτης θέσης κάθε γραμμής (i).

Στη **max** τελικά καταχωρίζεται η μεγαλύτερη από τις τιμές των θέσεων μνήμης της γραμμής **i**.

☞ Στο παραπάνω πρόγραμμα, οι μέγιστες τιμές κάθε γραμμής απλώς εμφανίζονται στην οθόνη. Αν θέλαμε να υπάρχουν και κάπου καταχωρισμένες, θα έπρεπε να χρησιμοποιήσουμε έναν πίνακα 100 θέσεων, π.χ max[100], όπου σε κάθε θέση μνήμης του πίνακα max[] θα αποθηκεύαμε τη μέγιστη τιμή της αντίστοιχης γραμμής του πίνακα a[].

Άσκηση 6

- Μια εταιρία κινητής τηλεφωνίας καταγράφει τις συνολικές χρεώσεις των πελατών της σε καθημερινή βάση για όλο το έτος. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει τις συνολικές χρεώσεις κάθε ημέρας και να τις καταχωρίζει σε έναν κατάλληλο πίνακα. Το πρόγραμμα θα πρέπει επίσης να εμφανίζει την ημέρα ή τις ημέρες (ημέρα και μήνα) με τη μεγαλύτερη χρέωση.

```
#include <iostream>
using namespace std;
int main()
{
	int xreosi[12][31],i,j,max;
	for (i=0;i<12;i++)
	{
		cout << "Μήνας "<<i+1<<endl;
		cout << "-----------------" << endl;
		for (j=0;j<31;j++)
		{
			cout << "Χρέωση "<<j+1<<"/"<<i+1<<":";
			cin >> xreosi[i][j];
		}
	}
	max=xreosi[0][0];
	for (i=0;i<12;i++)
		for (j=0;j<31;j++)
			if (xreosi[i][j]>max) max=xreosi[i][j];
	cout << " Ημέρες με τη μεγαλύτερη χρέωση " << max << endl;
	cout << "---------------------------------" <<  endl;
	for (i=0;i<12;i++)
	   for (j=0;j<31;j++)
			if (xreosi[i][j]==max) cout << j+1 << "/" << i+1 << endl;
   return 0;
}
```

Άσκηση 7

- Να γραφεί μια συνάρτηση με μια παράμετρο στην οποία να μπορεί να μεταβιβαστεί μια συμβολοσειρά. Κάθε φορά που θα καλείται η συνάρτηση, θα πρέπει να επιστρέφει ως τιμή το σύνολο των χαρακτήρων των συμβολοσειρών που της έχουν μεταβιβαστεί ως ορίσματα μέχρι εκείνη τη στιγμή. Για παράδειγμα, αν την πρώτη φορά κληθεί με το όρισμα «C++» θα πρέπει να επιστρέψει τη συμβολοσειρά «C++», αν την δεύτερη φορά κληθεί με το όρισμα «IS THE BEST» θα πρέπει να επιστρέψει τη συμβολοσειρά «C++ IS THE BEST», αν την τρίτη φορά κληθεί με το όρισμα «-------» θα πρέπει να επιστρέψει τη συμβολοσειρά «C++ IS THE BEST ------». Επίσης να γραφεί πρόγραμμα το οποίο θα ζητάει συνεχώς λέξεις από τον χρήστη και θα εμφανίζει το σύνολο των χαρακτήρων των συμβολοσειρών που έδωσε μέχρι στιγμής, χρησιμοποιώντας τη συνάρτηση που φτιάξατε. Η επαναληπτική λειτουργία του προγράμματος θα πρέπει να σταματάει όταν δοθεί η συμβολοσειρά «STOP»

```
#include <iostream>
using namespace std;
string all_strings(string lex);
int main()
{
		string fr;
		do
   {
      cin >> fr;
      cout << all_strings(fr) << endl;

   } while (fr!="STOP");
		return 0;
}
string all_strings(string lex)
{
		static string all="";
		all=all + lex;
		return all;
}
```

Άσκηση 8

- Να γραφεί ένα πρόγραμμα το οποίο θα ζητάει από τον χρήστη 20 ονόματα και θα τα καταχωρίζει σε έναν πίνακα αντικειμένων string 20 θέσεων. Το πρόγραμμα δεν θα πρέπει να επιτρέπει ην καταχώριση του ίδιου ονόματος για δεύτερη φορά. Να γράψετε και μια συνάρτηση που θα ελέγχει αν ένα όνομα έχει ήδη δοθεί.

```
#include <iostream>
#include <string>
using namespace std;

bool yparxei(string p[],int pos);

int main()

{
                string o[20];
                int i,j;
                for (i=0;i<20;i++)
   {
    do
    {
     cout << "Δώσε όνομα "<<i+1<<" ->";
     getline(cin,o[i]);
     if (yparxei(o,i))
        cout << "Το όνομα υπάρχει ήδη - δώστε άλλο"
<< endl;
     else
        break;
    } while (1);
   }
   return 0;
}
```

```
bool yparxei(string p[],int pos)
{
   bool brika=false;
   int j;
   for (j=0;j<pos-1;j++)
      if (p[j]==p[pos])    brika=true;
   return brika;
}
```

Άσκηση 9

- Να γραφεί ένα πρόγραμμα το οποίο να ζητάει από τον χρήστη να πληκτρολογήσει μια φράση. Κατόπιν να εμφανίζει στην οθόνη το πλήθος των χαρακτήρων διαστήματος που υπάρχουνε μέσα στη φράση. Να χρησιμοποιήσετε την κλάση **string**

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
	string frasi;
	int i=0,cnt=0;
	cout << "Δώσε μια φράση:";
	getline(cin,frasi);
	while (frasi[i]!='\0')
	{
		if (frasi[i]==' ') cnt++;
		i++;
	}
	cout << «ο χαρακτήρας του διαστήματος υπάρχει"<<cnt<<"
φορές"<<endl;
	return 0;
}
```