

Άσκηση 1

- Να γραφεί μια **υπερφορτωμένη** συνάρτηση με όνομα **myabs()**, η οποία θα δέχεται ως παράμετρο έναν αριθμό τύπου **sort int, int, float, double** ή **long double** και θα επιστρέφει ως τιμή την απόλυτη τιμή του αριθμού. Η τιμή που θα επιστρέφει πρέπει να είναι του ίδιου τύπου με την παράμετρο. Να γραφεί ένα πρόγραμμα που να καλεί όλες τις εκδόσεις της συνάρτησης.

```
#include <iostream>
using namespace std;
int myabs(int x)
{
    if (x<0) return -x;
}
short int myabs(short int x)
if (x<0) return -x;
}
float myabs(float x)
{
    if (x<0) return -x;
}
double myabs(double x)
{
    if (x<0) return -x;
}
long double myabs(long double x)
{
    if (x<0) return -x;
}
```

```
int main()
{
    short si=-12;
    int i=-100000;
    float f=-2.34;
    double d=-2.1234567;
    long double ld=-4.123456789;
    cout << myabs(si) << endl ;
    cout << myabs(i) << endl ;
    cout << myabs(f) << endl ;
    cout << myabs(d) << endl ;
    cout << myabs(ld) << endl ;
    return 0;
}
```

Άσκηση 2

- Να γραφεί ένα πρόγραμμα το οποίο θα ζητάει έναν ακέραιο και θα εξετάζει αν είναι πρώτος. Το πρόγραμμα θα εμφανίζει «Ο αριθμός είναι πρώτος» στη περίπτωση όπου ο έλεγχος είναι θετικός. Διαφορετικά θα εμφανίζει το μήνυμα «ο αριθμός δεν είναι πρώτος και έχει τους παρακάτω διαιρέτες:» και θα ακολουθεί η λίστα με τους διαιρέτες που εντοπίστηκαν.

```
#include <iostream>
using namespace std;
int main()
{
    int ar,i;
    bool protos=true;
    cout << "Δώσε αριθμό:";
    cin >> ar;
    for (i=2;i<ar/2;i++)
    {
        if (ar%i==0)
        {
            if (protos)
                cout << "Ο αριθμός δεν είναι πρώτος και έχει τους παρακάτω διαιρέτες:"<<endl;
            protos=false;
            cout << i << endl;
        }
    }
    if (protos)
        cout << "Ο αριθμός είναι πρώτος" << endl;
    return 0;
}
```

Άσκηση 3

- Να γραφεί ένα πρόγραμμα το οποίο θα ζητάει έναν ακέραιο θετικό αριθμό και θα εμφανίζει το άθροισμα των ψηφίων του. Πχ αν δώσεις 5234 θα εμφανίσει $5+2+3+4$

```
#include <iostream>
using namespace std;
int main()
{
    int ar,y,p,sum=0;
    cout << "Δώσε αριθμό:";
    cin >> ar;
    do
    {
        y=ar % 10;
        p=ar/10;
        sum=sum+y;
        ar=p;
    } while (p!=0);
    cout << "Το άθροισμα των ψηφίων είναι " << sum << endl;
    return 0;
}
```

Άσκηση 4

Να γραφεί μια συνάρτηση η οποία να διαθέτει 2 παραμέτρους. Η πρώτη τύπου **int** και η δεύτερη τύπου **string**. Η συνάρτηση πρέπει να εμφανίζει στην οθόνη τη συμβολοσειρά της δεύτερης παραμέτρου τόσες φορές όσο είναι η τιμή της πρώτης παραμέτρου. Στην περίπτωση που η συνάρτηση κληθεί χωρίς κανένα όρισμα, θα πρέπει να εμφανίζει το όνομα σας 10 φορές και στη περίπτωση που κληθεί με ένα μόνο όρισμα (έναν ακέραιο αριθμό) να εμφανίζει το όνομα σας τόσες φορές όσες η τιμή του ορίσματος.


```
#include <iostream>
using namespace std;
void display(int ar=10, string onom="Nikos")
{
    int i;
    for (i=1;i<=ar;i++)    cout << onom << endl;
}
int main()
{
    display();
    display(5);
    display(20,"Klearchos");
    return 0;
}
```

Άσκηση 5

- Θέλουμε να υπολογίσουμε το εμβαδόν της επιφάνειας ενός κύκλου, ενός ορθογωνίου παραλληλογράμμου και ενός ορθογωνίου παραλληλεπιπέδου. Ο υπολογισμός θα γίνεται από μια συνάρτηση με όνομα **emvadon** στην οποία θα μεταβιβάζονται οι διαστάσεις του σχήματος και αυτή θα επιστρέφει ως τιμή το εμβαδόν του. Η συνάρτηση θα πρέπει να **υπερφορτωθεί** 3 φορές ώστε όταν καλείται με ένα όρισμα να υπολογίζει το εμβαδόν του κύκλου, με 2 ορίσματα το εμβαδόν του παραλληλογράμμου και με 3 ορίσματα το εμβαδόν του παραλληλεπιπέδου. Να κατασκευαστεί ένα πρόγραμμα το οποίο να καλεί και τις 3 εκδόσεις της συνάρτησης ώστε να υπολογίζει και να εμφανίζει το εμβαδόν ενός κύκλου ακτίνας 20 μέτρων, το εμβαδόν ενός παραλληλογράμμου διαστάσεων 5x10 μέτρων και το εμβαδόν ενός ορθογωνίου παραλληλεπιπέδου διαστάσεων 4x6x8 μέτρων. Το π να δηλώνεται ως σταθερά 3.14159

```
#include <iostream>
#include <cmath>
#define pi 3.14159
using namespace std;
double emvadon(double r);
double emvadon(double a, double b);
double emvadon(double a, double b, double c);
int main()
{
    cout<<emvadon(20)<<endl;
    cout<<emvadon(5,10)<<endl;
    cout<<emvadon(4,6,8)<<endl;
    return 0;
}
double emvadon(double r)
{
    return pi*pow(r,2);
}
double emvadon(double a, double b)
{
    return a*b;
}
double emvadon(double a, double b, double c)
{
    return 2*(a*b+a*c+b*c);
}
```

Άσκηση 6

- Τροποποιήστε το προηγούμενο πρόγραμμα χωρίς να υπερφορτώσετε τη συνάρτηση **emvadoh** αλλά να επιτύχετε το ίδιο ακριβώς αποτέλεσμα χρησιμοποιώντας **προκαθορισμένες** τιμές παραμέτρων.

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#define pi 3.14159
using namespace std;
double emvadon(double a=0, double b=0, double c=0);
int main()
{
    cout<<emvadon(20)<<endl;
    cout<<emvadon(5,10)<<endl;
    cout<<emvadon(4,6,8)<<endl;
    return 0;
}
double emvadon(double a, double b, double c)
{
    if (a==0 && b==0 && c==0)
    {
        cout << " Λάθος" << endl;
        exit(1);
    }
    else if (b==0 && c==0)
    {
        cout << "Κύκλος:";
        return pi*pow(a,2);
    }
    else if (c==0)
    {
        cout << "Ορθογώνιο:";
        return a*b;
    }
    else
    {
        cout << "Παραλληλεπίπεδο:";
        return 2*(a*b+a*c+b*c);
    }
}
```