Εισαγωγή στον Προγραμματισμό

Ενότητα C++ (75%) :

6η εβδομάδα

Παναγιώτης Τζουνάκης

(βασισμένο στις σημειώσεις που ευγενικά προσέφερε ο καθηγητής κ. Γεώργιος Ραχώνης)

Άνοιξη 2025

Η πρόταση while

Η απλή μορφή

Σύνταξη

```
while (συνθήκη)
    πρόταση;
```

Η γενική μορφή

Σύνταξη

```
while (συνθήκη)
{
      πρόταση 1;
      πρόταση 2;
            .
            .
      πρόταση n;
}
```

Παράδειγμα W6.1

Να βρεθεί το παραγοντικό του φυσικού αριθμού *n* με χρήση της εντολής *while*

```
#include<iostream>
using namespace std;
int main( )
{
int n, a;
long int factorial;
cout<<“\nAssign value to n”;
cin>>n;
if (n==0)
        cout<<“\nThe factorial of 0 is 1”;
else
```

```
{
        factorial=1;
        a=1;
        while (a<=n)
        {
                factorial=factorial*a;
                a++;
        }
        cout<<“\nThe factorial of ”<<n<<“ is ”
                                <<factorial;
}
return 0;
}
```

Χρήση της πρότασης while
για έλεγχο ορθότητας δεδομένων

Παράδειγμα W6.2

Να βρεθεί το παραγοντικό του φυσικού αριθμού *n*

```
#include<iostream>
using namespace std;
int main( )
{
int n, a;
long int factorial;
cout<<“\nAssign value to n”;
cin>>n;
while(n<0)
{
    cout<<“\nThe value of n must be greater
                    than or equal to 0”;
    cout<<“\nAssign value to n”;
    cin>>n;
}
```

```
if (n==0)
        cout<<"\nThe factorial of 0 is 1";
else
{
        factorial=1;
        a=1;
        while (a<=n)
        {
                factorial=factorial*a;
                a++;
        }
        cout<<"\nThe factorial of n "<<n<<" is "
                        <<factorial;

}
return 0;
}
```

Η πρόταση do – while

Η απλή μορφή

Σύνταξη

```
do
   πρόταση;
while (συνθήκη);
```

Μπορείτε να διακρίνετε καμία διαφορά στην παραπάνω πρόταση και στην while;

Η γενική μορφή

Σύνταξη

```
do
{
    πρόταση 1;
    πρόταση 2;
              .
              .
    πρόταση n;
}
while (συνθήκη);
```

Παράδειγμα W6.3

Έστω ότι θέλουμε να βρούμε το μέγιστο κοινό διαιρέτη (ΜΚΔ) δύο θετικών ακεραίων αριθμών *α,β* με τη βοήθεια του τύπου

$$\mathrm{MK\Delta}(a,b)=\begin{cases}\mathrm{MK\Delta}(a-b,b) & \text{αν}\ \ \alpha>\beta\\ \mathrm{MK\Delta}(a,b-a) & \text{αν}\ \ \alpha<\beta\\ a & \text{αν}\ \ \alpha=\beta\end{cases}$$

Η παραπάνω διαδικασία θα πρέπει να επαναληφθεί ένα πλήθος φορών που δε γνωρίζουμε, έως ότου οι *α,β* να γίνουν ίσοι.

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,x,y;
    cout<<"Assign value to a:";
    cin>>a;
    cout<<"Assign value to b:";
    cin>>b;
    x=a;
    y=b;
    while(a!=b)
    {
        if (a>b)
        a=a-b;
```

```
    else
        b=b-a;
    }
    cout<<“The greatest common divisor of “
                <<x<<“ and ” <<y<<“  is ”<<a;
    return 0;
}
```

Παράδειγμα W6.4

Να γράψετε πρόγραμμα που θα εμφανίζει στην οθόνη όλους τους θετικούς ακέραιους των οποίων η τρίτη δύναμη είναι μικρότερη από τον αριθμό 0.1E300
Σημ: για να ελέγξετε την ορθότητα του προγράμματος σας, δοκιμάστε πρώτα με ένα μικρότερο όριο π.χ. 0.1E5

Χρήση της do - while για επανεκτέλεση του προγράμματος

Παράδειγμα W6.5

```
#include<iostream>
using namespace std;
int main()
{
        char restart;
        do
        {
                cout<<"This is an example"<<endl;
                cout<<"Do you want to continue (y/n)? ";
                cin>>restart;
        }
        while (restart=='y' ||restart=='Y');
        return 0;
}
```

Το πρόγραμμα εύρεσης του Μ.Κ.Δ. με επανεκτέλεση Παράδειγμα W6.5A

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,x,y;
    char restart;
    do
    {
        cout<<"Assign value to a:";
        cin>>a;
        cout<<"Assign value to b:";
        cin>>b;
        x=a;
        y=b;
```

```
        while(a!=b)
        {
            if (a>b)
            a=a-b;
        else
            b=b-a;
        }
        cout<<"The greatest common divisor of "
                    <<x<<" and " <<y<<"  is "<<a;
        cout<<"Do you want to continue (y/n)? ";
        cin>>restart;
    }
    while (restart=='y' ||restart=='Y');
    return 0;
}
```

Παράδειγμα W6.6

Η εκθετική συνάρτηση *exp(x)* ορίζεται ως το άθροισμα των άπειρων όρων της παρακάτω σειράς:

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^i}{i!} + \ldots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Να χρησιμοποιήσετε την παραπάνω σειρά για τον υπολογισμό του *exp(x)* έως ότου η τιμή του όρου

$$\left| \frac{x^i}{i!} \right|$$

γίνει μικρότερη ή ίση από μια επιθυμητή ακρίβεια *A*.

Παράδειγμα W6.7

Να γραφεί πρόγραμμα που θα υπολογίζει τις τιμές της συνάρτησης *sin(x)* που δίνεται από τον τύπο

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

έως ότου (η προσέγγιση *A* θα δίνεται από το χρήστη):

$$\left| (-1)^n \frac{x^{2n+1}}{(2n+1)!} \right| < A$$

Παράδειγμα W6.8

Να γραφεί πρόγραμμα που θα υπολογίζει τον αριθμό *π* με ακρίβεια *A* που θα δίνει ο χρήστης, με την βοήθεια του αναπτύγματος (Euler 1730):

$$\pi = \sqrt{6\left(1+\frac{1}{4}+\frac{1}{9}+\ldots+\frac{1}{n^2}+\ldots\right)}$$

Άσκηση W6_E1

Να γράψετε πρόγραμμα που θα εμφανίζει στην οθόνη όλους τους θετικούς ακέραιους των οποίων το παραγοντικό είναι μικρότερο από τον αριθμό 0.1E303

Άσκηση W6.1

Να γράψετε πρόγραμμα που θα χρησιμοποιεί τον τύπο του S. Plouffe (1995)

$$\pi = \sum_{i=0}^{\infty} \frac{1}{16^i} \left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6} \right)$$

για τον υπολογισμό του π,

α) με εύρεση του αθροίσματος έως τον 1000 όρο

β) ζητώντας από το χρήστη να δώσει ένα «μικρό» αριθμό a και υπολογίζοντας το παραπάνω άθροισμα έως ότου ο i όρος να γίνει < a.

ΠαράδειγμαW6.9

Να γράψετε πρόγραμμα που θα χρησιμοποιεί την σειρά των Gregory-Leipniz

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \ldots = \sum_{i=0}^{\infty} 4\frac{(-1)^i}{2i+1}$$

για τον υπολογισμό του π,

α) με εύρεση του αθροίσματος έως τον 1000 όρο

β) ζητώντας από το χρήστη να δώσει ένα «μικρό» αριθμό a και υπολογίζοντας το παραπάνω άθροισμα έως ότου ο i όρος να γίνει < a.

Άσκηση W6.2

Να γράψετε πρόγραμμα που θα χρησιμοποιεί την σειρά

$$\pi = 3 + \frac{4}{2\times3\times4} - \frac{4}{4\times5\times6} + \frac{4}{6\times7\times8} - \frac{4}{8\times9\times10} + \ldots$$

για τον υπολογισμό του π,

α) με εύρεση του αθροίσματος έως τον 1000 όρο

β) ζητώντας από το χρήστη να δώσει ένα «μικρό» αριθμό a και υπολογίζοντας το παραπάνω άθροισμα έως ότου ο i όρος να γίνει < a.

ΠαράδειγμαW6.10

Να γράψετε πρόγραμμα που θα διαβάζει *n* ακέραιους αριθμούς (το *n* θα το δίνει ο χρήστης) και στη συνέχεια θα τυπώνει:

α) το πλήθος των αριθμών που βρίσκονται στο διάστημα [-1000,3000],

β) το γινόμενο των αριθμών που είναι διάφοροι του μηδενός και βρίσκονται εκτός του διαστήματος [-50,500],

γ) το πλήθος και το άθροισμα των αριθμών που είναι πολλαπλάσια του 5,

δ) τον μέσο όρο των άρτιων αριθμών που βρίσκονται έξω από το διάστημα [3,9000].

Άσκηση W6_E2

Να γράψετε πρόγραμμα που θα ζητάει από το χρήστη μη αρνητικούς ακέραιους (απαιτείται έλεγχος ορθότητας δεδομένων) και θα εμφανίζει στην οθόνη εκείνους των οποίων το παραγοντικό είναι αριθμός μικρότερος από το 0.1E303. Το πρόγραμμα θα σταματάει όταν ο χρήστης δώσει σαν είσοδο το 0.

Άσκηση W6_E3

Να γράψετε πρόγραμμα που θα εμφανίζει στην οθόνη όλους τους θετικούς ακέραιους των οποίων η παραγοντικό βρίσκεται στο διάστημα [0.3E100, 0.5E300].