

Εργαστήριο 6:

6.1 Δομές Επανάληψης – Βρόγχοι (Loops)

Όταν θέλουμε να επαναληφθεί μια ομάδα εντολών τη βάζουμε μέσα σε ένα βρόχο επανάληψης. Το αν θα (ξανα)επαναληφθεί η εκτέλεση της ομάδας εντολών καθορίζεται από το αν ισχύει μια συγκεκριμένη συνθήκη.

Στη C++ υπάρχουν τρεις τύποι βρόχων: (a) while, (b) do while, και (c) for. Ακολουθεί η σύνταξη για κάθε μια:

```
while (συνθήκη) {ακολουθία_εντολών}
```

Αν ισχύει η συνθήκη, θα εκτελεστεί η ακολουθία_εντολών και για όσο ισχύει η συνθήκη, θα ξαναεκτελείται η ακολουθία_εντολών. Αν η συνθήκη αρχικά είναι ψευδής, η ακολουθία_εντολών δεν θα τρέξει καθόλου.

```
do {ακολουθία_εντολών} while (συνθήκη);
```

Εκτελείται η ακολουθία_εντολών και αν και για όσο ισχύει η συνθήκη, θα ξαναεκτελείται η ακολουθία_εντολών. Δηλαδή η ακολουθία_εντολών θα εκτελεστεί τουλάχιστον μια φορά.

```
for (ανάθεση_αρχικοποίηση ; συνθήκη ; ανάθεση_μεταβολή) {ακολουθία_εντολών}
```

Εκτελείται μία μόνο φορά η ανάθεση_αρχικοποίηση (συνήθως δίνει αρχική τιμή σε μια μεταβλητή που λέγεται μεταβλητή ελέγχου). Στη συνέχεια, αν ισχύει η συνθήκη, θα εκτελεστεί η ακολουθία_εντολών και μετά η ανάθεση_μεταβολή, η οποία συνήθως αυξάνει ή μειώνει τη μεταβλητή ελέγχου κατά ένα σταθερό "βήμα". Για όσο ισχύει η συνθήκη, θα ξαναεκτελείται η ακολουθία_εντολών και ακολούθως η ανάθεση_μεταβολή.

ΠΑΡΑΤΗΡΗΣΕΙΣ:

Όπως και στην περίπτωση της if, αν η ακολουθία εντολών είναι μόνο μια εντολή, τα άγκιστρα είναι προαιρετικά. Συνίσταται όμως σε αρχάριους προγραμματιστές να χρησιμοποιούν πάντα άγκιστρα. Η παρένθεση στην πρόταση είναι υποχρεωτική.

Προσοχή στο ';' της do { ... } while (...); Η for και η while δεν τελειώνουν με ';

Ο βρόχος for χρησιμοποιείται συνήθως όταν το πρόγραμμα γνωρίζει από πριν πόσες φορές θα εκτελεστεί το μπλοκ εντολών. Απλό παράδειγμα:

```
for (i=10 ; i<100 ; i++)  
{ cout<< i << "\t";}
```

Εκτυπώνει όλους τους διψήφιους θετικούς ακεραίους. Ο βρόχος for χρησιμοποιείται συνήθως όταν ο αριθμός των επαναλήψεων είναι γνωστός πριν την είσοδο στο βρόχο. Η πρόταση του βρόχου που βρίσκεται μέσα στην παρένθεση αποτελείται από τρία ορίσματα χωρισμένα με ερωτηματικό.

1. Το πρώτο εκτελείται πριν την είσοδο στο βρόχο. Συνήθως δίνει μια αρχική τιμή σε μια βοηθητική μεταβλητή.

2. Το δεύτερο είναι μια συνθήκη. Ο βρόχος τρέχει για όσο η συνθήκη ικανοποιείται (είναι αληθής). Συνήθως η συνθήκη ελέγχει αν η τιμή της μεταβλητής βρίσκεται στο επιθυμητό διάστημα.

3. Το τρίτο είναι μια εντολή, η οποία τρέχει κάθε φορά που τελειώνει το σώμα των εντολών του βρόχου και συνήθως εκφράζει το βήμα με το οποίο μεταβάλλεται η μεταβλητή.

Καθένα από τα 3 αυτά ορίσματα στην πρόταση του βρόχου for συνήθως κάνει ένα πράγμα, ωστόσο μπορεί να είναι και κενό. Π.χ. κενή τρίτη εντολή σημαίνει ότι δεν υπάρχει μεταβολή του μετρητή. Αυτό προκαλεί την επ' άπειρο εκτέλεση του βρόχου, εκτός αν διακοπεί με κάποιο άλλο τρόπο, π.χ. με μια εντολή break.

Είναι επίσης δυνατό να χρησιμοποιήσουμε πολλές μεταβλητές μαζί στην πρώτη ή στην τρίτη θέση, χωρίζοντάς τες με κόμμα. Αυτό επιτρέπει στο βρόχο να ελέγχει περισσότερες από μία μεταβλητές. Στο παρακάτω παράδειγμα φαίνεται ο ορισμός ενός τέτοιου βρόχου, με τις μεταβλητές hi και lo να ξεκινούν από το 100 και το 0 αντιστοίχως και να τείνουν στο ίδιο σημείο.

```
for (hi = 100, lo = 0; hi >= lo; hi--, lo++)
```

Παράδειγμα 1:

Υπολογίστε το άθροισμα της παράστασης:

$1/2 + 1/4 + 1/6 + 1/8 + \dots + 1/MAX$, όπου MAX έχει ορισθεί ως σταθερά που παίρνει την τιμή 20.

Στο παράδειγμα αυτό ο αριθμός των επαναλήψεων είναι συγκεκριμένος και η βοηθητική μεταβλητή i που θα χρησιμοποιήσουμε πρέπει να παίρνει τιμές από 2 έως και 20 με βήμα 2. Προφανώς δηλαδή θα γίνουν 10 επαναλήψεις. Για να δηλώσω μια σταθερά είτε το κάνω με το define όπως φαίνεται στη λύση του παραδείγματος 1 είτε το κάνω με τη λέξη κλειδί const. Το αποτέλεσμα του αθροίσματος σε κάθε επανάληψη (στο τέλος κάθε loop) τυπώνεται στην οθόνη (αν μετρήσετε τις φορές που τυπώνεται “ το athroisma einai ” είναι 10 όσες και οι επαναλήψεις) ενώ το συνολικό άθροισμα τυπώνεται στο τέλος της άσκησης έξω από το for.

```

#include <iostream>
using namespace std;

#define MAX 20

int main()
{
    int i;
    float sum = 0;
    for (i = 2 ; i <= 20; i = i +2)
    {
        sum = sum + 1.0/i;
        cout <<"to athroisma einai " << sum <<endl;
    }

    cout <<"to teliko athroisma einai " << sum <<endl;

    system("pause");
    return 0;
}

```

```

C:\Users\Giannis\Desktop\C_arrays\test1.exe
to athroisma einai 0.5
to athroisma einai 0.75
to athroisma einai 0.916667
to athroisma einai 1.04167
to athroisma einai 1.14167
to athroisma einai 1.225
to athroisma einai 1.29643
to athroisma einai 1.35893
to athroisma einai 1.41448
to athroisma einai 1.46448
to teliko athroisma einai 1.46448
Press any key to continue . . .

```

Παράδειγμα 2:

Υπολογίστε το άθροισμα της παράστασης:

$$1/2 + 1/4 + 1/6 + 1/8 + \dots ,$$

για τους MAX πρώτους όρους όπου το MAX έχει ορισθεί ως σταθερά που παίρνει την τιμή 20.

Η άσκηση διαφέρει από την προηγούμενη, μας ζητάει να υπολογίσουμε το άθροισμα για τους MAX πρώτους όρους, δηλαδή για 20 όρους αφού MAX=20. Άρα πρέπει να κάνουμε MAX επαναλήψεις.

```

#include <iostream>
using namespace std;

int main()
{
    int i, paranomastis=1;
    const int MAX = 20;
    float sum =0;

    for (i = 1; i <= MAX; i++)
    {
        paranomastis = paranomastis*2;
        sum = sum + 1.0/paranomastis;
        cout<< "To athroisma einai " << sum <<endl;
        cout << " kai prosthesame ton " << i <<" oro " << endl;
    }
    cout<< "To teliko athroisma einai " << sum <<endl;

    system ("pause");
    return 0;
}

```

```

To athroisma einai 0.5
kai prosthesame ton 1 oro
To athroisma einai 0.75
kai prosthesame ton 2 oro
To athroisma einai 0.875
kai prosthesame ton 3 oro
To athroisma einai 0.9375
kai prosthesame ton 4 oro
To athroisma einai 0.96875
kai prosthesame ton 5 oro
To athroisma einai 0.984375
kai prosthesame ton 6 oro
To athroisma einai 0.992188
kai prosthesame ton 7 oro
To athroisma einai 0.996094
kai prosthesame ton 8 oro
To athroisma einai 0.998047
kai prosthesame ton 9 oro
To athroisma einai 0.999023
kai prosthesame ton 10 oro
To athroisma einai 0.999512
kai prosthesame ton 11 oro
To athroisma einai 0.999756
kai prosthesame ton 12 oro
To athroisma einai 0.999878
kai prosthesame ton 13 oro
To athroisma einai 0.999939
kai prosthesame ton 14 oro
To athroisma einai 0.999969
kai prosthesame ton 15 oro
To athroisma einai 0.999985
kai prosthesame ton 16 oro
To athroisma einai 0.999992
kai prosthesame ton 17 oro
To athroisma einai 0.999996
kai prosthesame ton 18 oro
To athroisma einai 0.999998
kai prosthesame ton 19 oro
To athroisma einai 0.999999
kai prosthesame ton 20 oro
To teliko athroisma einai 0.999999
Press any key to continue . . . _

```

6.2 Μετρητές – Αθροιστές – Πολλαπλασιαστές

Παράδειγμα 3:

Να υλοποιηθεί πρόγραμμα σε C++ στο οποίο ο χρήστης να εισάγει πραγματικούς αριθμούς και το πρόγραμμα να βρίσκει το άθροισμα, το γινόμενο και το πλήθος των αριθμών αυτών. Η εισαγωγή αριθμών θα σταματήσει όταν ο χρήστης θα δώσει 0 ή 1, οπότε το πρόγραμμα θα εκτυπώσει τα αποτελέσματα και θα τερματίσει. (Δηλ. το 0 ή το 1 που θα δώσει για τον τερματισμό να μη ληφθούν υπόψη στο άθροισμα, το γινόμενο και την καταμέτρηση).

Τρόπος Σκέψης:

Για τη μέτρηση των αριθμών θα χρησιμοποιήσουμε μετρητή, δηλ μια μεταβλητή που θα αρχικοποιείται με μηδενική τιμή και κάθε φορά που θα εισάγεται ένας έγκυρος αριθμός, ο μετρητής θα αυξάνει κατά ένα.

Για το άθροισμα θα χρησιμοποιήσουμε αθροιστή, δηλ μια μεταβλητή που θα αρχικοποιείται με μηδενική τιμή και κάθε φορά που θα εισάγεται ένας έγκυρος αριθμός ο αριθμός αυτός θα προστίθεται στον αθροιστή.

Για το γινόμενο θα χρησιμοποιήσουμε πολλαπλασιαστή, δηλ. μια μεταβλητή που θα αρχικοποιείται στη μονάδα και κάθε φορά που θα εισάγεται ένας έγκυρος αριθμός ο αριθμός αυτός θα πολλαπλασιάζεται με τον πολλαπλασιαστή και το αποτέλεσμα θα αποθηκεύεται στον πολλαπλασιαστή.

Για τον τύπο του βρόχου σκεφτόμαστε ως εξής:

Δεν γνωρίζουμε τον αριθμό των επαναλήψεων άρα δεν χρησιμοποιούμε for. Άρα θα χρησιμοποιήσουμε while – do ή do - while, δηλ. για όσο ο χρήστης δεν δίνει 0 ούτε δίνει 1, να εκτελούνται οι 3 πράξεις.

ΠΡΟΣΟΧΗ:

Η συνθήκη για να σταματήσει το πρόγραμμα όταν δοθεί ο αριθμός 1 ή ο αριθμός 0 είναι η:

```
while ( (a!=0) && (a!=1))
```

όπου a ο αριθμός που δίνεται από το πληκτρολόγιο.

Δηλαδή παρόλο που γράφουμε πως θα σταματήσει όταν δοθεί είτε ο ένας είτε ο άλλος αριθμός επειδή το while μεταφράζεται με «όσο» η συνθήκη που πρέπει να γράψουμε είναι για πόσο θα συνεχίζονται οι επαναλήψεις. Δηλαδή πρέπει να γράψουμε ότι οι επαναλήψεις συνεχίζονται όσο το a είναι διάφορο του 0 ΚΑΙ διάφορο του 1.

```

#include <iostream>
using namespace std;

int main()
{
    int counter = 0;
    float a, sum =0, product =1;

    cout << "Dwse a pragmatiko " <<endl;
    cin >> a;

    while ((a !=0) && (a!=1))
    {
        counter = counter + 1;
        sum = sum + a ;
        product = product * a;
        cout<< "Dwse kai allo pramgatiko " <<endl;
        cin >> a;
    }
    cout << "Dothikan synolika "<< counter <<" arithmoi " <<endl;
    cout<< "To teliko athroisma einai " << sum <<endl;
    cout<< "To teliko ginomeno einai " << product <<endl;
    system ("pause");
    return 0;
}

```

6.3 Ασκήσεις

Άσκηση 6.1. Να τυπωθούν οι αριθμοί από το 101 έως το 200 χρησιμοποιώντας και τις 3 δομές επανάληψης. Ζητούμενο επίσης της άσκησης είναι να τυπώνονται σε κάθε γραμμή της οθόνης 10 αριθμοί και μεταξύ τους να υπάρχει ένα κενό ('\t').

Άσκηση 6.2. Να φτιαχτεί πρόγραμμα το οποίο θα μετρά πόσους θετικούς και αρνητικούς ακέραιους έχει δώσει ένας χρήστης (φτιάχνετε δηλαδή μια δομή επανάληψης που θα ζητάει από το χρήστη ακέραιους, θα ξεχωρίζει θετικούς και αρνητικούς και θα μετράει πόσοι θετικοί δόθηκαν και πόσοι αρνητικοί). Το πρόγραμμα θα σταματά όταν δοθεί ο αριθμός 10 από το πληκτρολόγιο. Υπολογίστε επίσης το άθροισμα των αριθμών, το γινόμενο και το μέσο όρο τους.

Άσκηση 6.3. Να υπολογιστεί χρησιμοποιώντας και τις 3 δομές επανάληψης το άθροισμα της συνάρτησης $1+1/2+1/3+1/4+.....+1/100$ για τους πρώτους 100 θετικούς ακέραιους.

Άσκηση 6.4. Φτιάξτε ένα πρόγραμμα που θα ζητάει από το χρήστη (τιμές από το πληκτρολόγιο) 5 ακέραιους (χρησιμοποιώντας δομή επανάληψης) και θα τυπώνει στην οθόνη σε ξεχωριστές γραμμές, τον αριθμό που δόθηκε και το τετραπλάσιο του.

Άσκηση 6.5. Χρησιμοποιήστε τις 3 δομές επανάληψης για να υπολογίσετε το άθροισμα της παράστασης $10+20+ \dots+90 + 100$. Τυπώστε τα αποτελέσματα για κάθε ζεύγος i και sum καθώς και το τελικό άθροισμα.

Άσκηση 6.6. Χρησιμοποιώντας κατάλληλη δομή επανάληψης να βρείτε πόσοι αριθμοί χρειάζονται ώστε το άθροισμα $1+1/2+1/3+1/4+\dots$ να γίνει μεγαλύτερο του 8.

Άσκηση 6.7. Να υλοποιηθεί αλγόριθμος σε C++ που να υπολογίζει τις παραστάσεις $y=(x^2-1)/2$ όπου x παίρνει τιμές όλους τους ζυγούς αριθμούς από το 300 ως το 150.

Άσκηση 6.8. Να υπολογιστεί για τους πρώτους 1000 θετικούς αριθμούς το άθροισμα

$$1+2/3+3/5+4/7 + \dots +1000/n .$$

Άσκηση 6.9. Δημιουργήστε προγράμματα τα οποία στην οθόνη θα έχουν την ακόλουθη έξοδο:

**

*

Τροποποιήστε το πρόγραμμα κατάλληλα ώστε να ξεκινάμε να τυπώνουμε από 1 αστεράκι μέχρι τα 6.

Τροποποιήστε το πρόγραμμα κατάλληλα ώστε να τυπωθούν 6 αστεράκια σε 6 γραμμές.

Άσκηση 6.10. Δημιουργήστε ένα πρόγραμμα το οποίο θα δέχεται σαν παραμέτρους (θα παίρνουν τιμές από το πληκτρολόγιο) α) τον μέγιστο αριθμό των χαρακτήρων που θα τυπωθεί στην οθόνη και b) το σύμβολο που θα τυπωθεί.

Για παράδειγμα αν ο μέγιστος αριθμός είναι 5 και το σύμβολο είναι @ τότε το πρόγραμμα θα μας τυπώνει τα εξής:

@@@@@

@@@@

@@@

@@

@