

Εισαγωγή στον Προγραμματισμό με C, C++

Εντολές Επανάληψης

Ιωάννης Τσιμπερίδης
itsimper@cs.duth.gr

Δομή Επανάληψης

Όταν ένα τμήμα κώδικα απαιτείται να εκτελεστεί αρκετές φορές, ώστε να βρεθεί η λύση ενός προβλήματος, τότε χρησιμοποιείται η **Δομή Επανάληψης**, κατά την οποία μία ή περισσότερες εντολές εκτελούνται επαναληπτικά, για ένα πλήθος φορών.

Το πόσες φορές θα εκτελεστούν αυτές οι εντολές μπορεί να είναι εκ των προτέρων γνωστό, ή όχι, και ανάλογα χρησιμοποιείται διαφορετική τακτική και εντολή.

Η C, όπως και πολλές άλλες γλώσσες προγραμματισμού, διαθέτει εντολές που υλοποιούν τη Δομή Επανάληψης, και συγκεκριμένα τις εντολές **“while”**, **“do...while”**, και **“for”**.

Στις εντολές **“while”** και **“do...while”** δεν είναι γνωστό εκ των προτέρων το πόσες φορές θα επαναληφθεί η εκτέλεση των εντολών. Αντίθετα, στη **“for”** είναι γνωστό.

Εντολή “while”

Συντάσσεται ως εξής:

```
while ((συνθήκη)) {  
    Εντολές  
}
```

Όπου (συνθήκη) είναι μία παράσταση που το αποτέλεσμα της μπορεί να είναι μηδέν, που αντιστοιχίζεται στο Ψευδές, ή μη μηδενική ποσότητα, που αντιστοιχίζεται στο Αληθές.

Στην περίπτωση που η (συνθήκη) είναι Αληθής, τότε εκτελούνται οι Εντολές και στη συνέχεια η (συνθήκη) ξαναεξετάζεται. Στην περίπτωση που είναι Ψευδής, η εκτέλεση του προγράμματος συνεχίζεται μετά το άγκιστρο κλεισίματος (}).

Εάν οι Εντολές είναι μόνο μία, τότε τα άγκιστρα είναι περιττά.

Εντολή “do...while”

Παρόμοια με την “while” είναι η “do... while”, η οποία συντάσσεται ως εξής:

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

Εκτελούνται οι Εντολές και στη συνέχεια ελέγχεται η (συνθήκη). Εάν είναι Αληθής, οι Εντολές εκτελούνται ακόμα μία φορά, κοκ. Εάν είναι Ψευδής το πρόγραμμα συνεχίζει μετά την εντολή “while”.

Οι διαφορές της “do...while” με τη “while” είναι ότι η (συνθήκη) ελέγχεται στο τέλος, με συνέπεια ο βρόχος να εκτελείται τουλάχιστον μία φορά, και ότι στο κλείσιμο της δομής απαιτείται το ελληνικό ερωτηματικό (;).

Εντολή “for”

Η εντολή “for” συντάσσεται ως εξής:

```
for ( (αρχ_μεταβ) ; (συνθήκη) ; (αλλ_μεταβ) ) {  
    Εντολές  
}
```

όπου (αρχ_μεταβ) είναι μία Μεταβλητή η οποία παίρνει αρχική τιμή και παίζει το ρόλο του μετρητή στις επαναλήψεις. Όπου (συνθήκη) είναι η λογική παράσταση η οποία όσο είναι Αληθής επαναλαμβάνεται ο βρόχος. Όπου (αλλ_μεταβ) είναι ο τρόπος με τον οποίο αλλάζει η Μεταβλητή-δείκτης ύστερα από κάθε επανάληψη.

Η αλλαγή της Μεταβλητής-δείκτη γίνεται αυτόματα και δεν χρειάζεται να υπάρχει εντολή που θα την αλλάζει εντός του βρόχου, όχι όμως ότι είναι απαγορευτικό.

Εάν οι Εντολές είναι μία, τα άγκιστρα είναι περιττά.

Εμφώλευση

Οι εντολές “if”, “switch”, “while”, “do...while”, και “for” επιτρέπουν ή όχι την εκτέλεση ενός σώματος εντολών, ή το εκτελούν ένα πλήθος φορές. Μία ή περισσότερες από τις εντολές αυτού του σώματος μπορεί να είναι και πάλι “if”, “switch”, “while”, “do...while”, ή “for”. Αυτό καλείται **εμφώλευση**, κατά την οποία μία εντολή επιλογής ή επανάληψης μπορεί να βρίσκεται εντός μίας άλλης εντολής επιλογής ή επανάληψης, και είναι ένα από τα χαρακτηριστικά του προγραμματισμού που επεκτείνει τις δυνατότητές του.

Στη C δεν υπάρχει κάποιο όριο στο βάθος των εμφωλεύσεων, δηλαδή εμφώλευση μέσα σε εμφώλευση που είναι μέσα σε εμφώλευση, κοκ. Ωστόσο, οι εμφωλεύσεις αυξάνουν την πολυπλοκότητα μεταγλώττισης και εκτέλεσης ενός προγράμματος, θέτοντας έτσι τεχνικά όρια.

Για παράδειγμα, μία “if” μπορεί να βρίσκεται εντός μίας “while”, που βρίσκεται εντός μίας από τις “case” μιας “switch”, που με τη σειρά της βρίσκεται εντός μίας “for”.

Παραδείγματα 1

Να υπολογιστεί το άθροισμα των αριθμών από το 1 έως και το 100.

```
int sum=0,i;
for (i=1;i<=100;i++)
    sum+=i;
printf("sum=%d",sum);
```

Να εμφανιστούν τα τετράγωνα των αριθμών από 0 έως 1, ανά 0.1.

```
double x;
for (x=0;x<=1;x+=0.1)
    printf("%lf\n",x*x);
```

Παραδείγματα 2

Να βρεθεί ο μεγαλύτερος και ο μικρότερος από τους ακέραιους αριθμούς που δίνονται, μέχρι να δοθεί το 0.

```
int a,min,max;
printf("Give an integer: "); scanf("%d",&a);
min=a; max=a;
while (a!=0) {
    if (a>max) max=a;
    else if (a<min) min=a;
    printf("Give another integer: "); scanf("%d",&a);
}
printf("max=%d\nmin=%d",max,min);
```