

Επιλογή Εντολών προς Εκτέλεση

ή

Διακλάδωση Προγράμματος

Μια βασική ιδιότητα προγραμματισμού έγκειται στην ικανότητά να λαμβάνει αποφάσεις. Είδατε πώς αυτές οι δυνατότητες λήψης αποφάσεων χρησιμοποιήθηκαν στην εκτέλεση των διαφόρων εντολών βρόχων για να καθοριστεί πότε θα τερματιστεί ο βρόχος του προγράμματος. Χωρίς τέτοιες δυνατότητες, δεν θα μπορούσε ποτέ να τερματιστεί ένας βρόχος και θα κατέληγε να εκτελεί την ίδια ακολουθία εντολών ξανά και ξανά, θεωρητικά για πάντα (γι 'αυτό ένας τέτοιος βρόχος προγράμματος ονομάζεται άπειρος βρόχος).

Η γλώσσα προγραμματισμού C παρέχει εντολές λήψης αποφάσεων ή διακλάδωσης :

- εντολή if
- εντολή switch

Η Εντολή if

```
if (παράσταση) {  
    εντολές  
}
```

```
// Εύρεση απόλυτης τιμής αριθμού - lecture04_1
```

```
#include <stdio.h>
```

```
int main(){
```

```
    int number;
```

```
    printf("Δώσε έναν αριθμό:");
```

```
    scanf("%i",&number);
```

```
    if (number < 0){
```

```
        number = -number;
```

```
    }
```

```
    printf("Η απόλυτη τιμή του αριθμού είναι: %i \n",number);
```

```
    return 0;
```

```
}
```

```
/*Πρόγραμμα υπολογισμού του μέσου όρου ενός συνόλου βαθμολογιών  
και του αριθμού των μαθημάτων κάτω από τη βάση - lecture04_2 */  
#include <stdio.h>
```

```
#define BASE 5
```

```
int main (void){  
    int numberOfGrades, grade;  
    int gradeTotal = 0;  
    int failureCount = 0;  
    float average;  
  
    printf ("Πόσους βαθμούς θα εισάγεις ; ");  
    scanf ("%i", &numberOfGrades);  
    for (int i = 1; i <= numberOfGrades; ++i ) {  
        printf ("Δώσε βαθμό #%i: ", i);  
        scanf ("%i", &grade);  
        gradeTotal = gradeTotal + grade;  
        if ( grade < BASE ){  
            failureCount++;  
        }  
    }  
    average = (float) gradeTotal / numberOfGrades;  
    printf ("\nΜέσος όρος = %.2f\n", average);  
    printf ("Μαθήματα κάτω από τη βάση = %i\n", failureCount);  
    return 0;  
}
```

- #define BASE 5 – Magic Numbers !!
- Δήλωση int i μέσα στο for
- Αν $average = gradeTotal / numberOfGrades$; τότε ακέραια διαίρεση και το αποτέλεσμα θα είναι ακέραιος !!
- %.2f

Πόσους βαθμούς θα εισάγεις ; 0

Μέσος όρος = -nan

Μαθήματα κάτω από τη βάση = 0

Η δομή if – else

//Πρόγραμμα ελέγχου αρτιότητας αριθμού

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int number_to_test, remainder;
```

```
    printf ("Δώσε αριθμό για έλεγχο: ");
```

```
    scanf ("%i", &number_to_test);
```

```
    remainder = number_to_test % 2;
```

```
    if ( remainder == 0 ){
```

```
        printf ("Ο αριθμός είναι άρτιος.\n");
```

```
    }
```

```
    if ( remainder != 0 ){
```

```
        printf ("Ο αριθμός είναι περιττός.\n");
```

```
    }
```

```
    return 0;
```

```
}
```

Μόνο το ένα από τα δύο if θα είναι TRUE

```
if (παράσταση) {
```

```
    εντολές
```

```
}
```

```
else {
```

```
    εντολές
```

```
}
```

```
//Πρόγραμμα ελέγχου αρτιότητας αριθμού
#include <stdio.h>
int main (void)
{
    int number_to_test, remainder;
    printf ("Δώσε αριθμό για έλεγχο: ");
    scanf ("%i", &number_to_test);
    remainder = number_to_test % 2;
    if ( remainder == 0 ){
        printf ("Ο αριθμός είναι άρτιος.\n");
    }
    else {
        printf ("Ο αριθμός είναι περιττός.\n");
    }
    return 0;
}
```

Σύνθετοι Σχεσιακοί Έλεγχοι

Έστω ότι εισάγουμε σε ένα πρόγραμμα βαθμούς διαγωνισμάτων και θέλουμε:

- Να ελέγχουμε αν οι βαθμοί εντός των ορίων δηλαδή ≥ 0 και ≤ 10 .
- Να έχουμε τον αριθμό των βαθμών που είναι μεταξύ 6 και 8.

Μπορούμε να χρησιμοποιήσουμε πολλαπλά if για κάθε περίπτωση. Η C όμως μας δίνει τη δυνατότητα να το υλοποιήσουμε ευκολότερα με σύνθετους ελέγχους.

Για το όριο των βαθμών:

```
if ( grade < 0 || grade > 10){
    printf("Βαθμός εκτός ορίων");
}
```

Παράσταση || Παράσταση

TRUE		TRUE	→	TRUE
TRUE		FALSE	→	TRUE
FALSE		TRUE	→	TRUE
FALSE		FALSE	→	FALSE

Για τη δεύτερη απαίτηση:

```
if ( grade >= 6 && grade <= 8){  
    number_of_grades++;  
}
```

Παράσταση && Παράσταση

TRUE	&&	TRUE	→	TRUE
TRUE	&&	FALSE	→	FALSE
FALSE	&&	TRUE	→	FALSE
FALSE	&&	FALSE	→	FALSE

Δίσεκτα έτη

Για να προσδιορίσουμε αν ένα έτος είναι δίσεκτο εφαρμόζουμε τα εξής:

1. Ελέγχουμε το υπόλοιπο της ακέραιας διαίρεσης του έτους με το 4. Αν είναι μηδέν ελέγχουμε το υπόλοιπο της ακέραιας διαίρεσης του έτους με το 100. Αν αυτό το υπόλοιπο είναι διαφορετικό του μηδενός τότε το έτος είναι δίσεκτο.
2. Αν από τον έλεγχο 1 δεν προκύψει ότι το έτος είναι δίσεκτο ελέγχουμε το υπόλοιπο της ακέραιας διαίρεσης του έτους με το 400. Αν είναι μηδέν τότε το έτος είναι δίσεκτο, άσχετα από το αποτέλεσμα του ελέγχου 1.

//Πρόγραμμα προσδιορισμού δίσεκτου έτους – lecture04_5.c

```
#include <stdio.h>
```

```
int main (void) {  
    int year;  
    printf ("Δήλωσε το έτος για έλεγχο: ");  
    scanf ("%i", &year);  
    if ( (year%4 == 0 && year%100 != 0) || year%400 == 0 ){  
        printf ("Είναι δίσεκτο έτος.\n");  
    }  
    else {  
        printf ("Δεν είναι δίσεκτο έτος.\n");  
    }  
    return 0;  
}
```

else if

```
if ( παράσταση 1 ) {  
    εντολές 1  
}  
else if ( παράσταση 2 ) {  
    εντολές 2  
}  
else {  
    εντολές 3  
}
```

// Πρόγραμμα υλοποίησης συνάρτησης προσήμου

```
#include <stdio.h>  
  
int main (void) {  
    int number, sign;  
  
    printf ("Please type in a number: ");  
    scanf ("%i", &number);  
    if ( number < 0 ){  
        sign = -1;  
    }  
    else if ( number == 0 ){  
        sign = 0;  
    }  
    else {  
        // Πρέπει να είναι θετικός  
        sign = 1;  
    }  
    printf ("Sign = %i\n", sign);  
    return 0;  
}
```

```
/* Πρόγραμμα υπολογισμού απλών παραστάσεων - τιμή σύμβολο πράξης τιμή */
#include <stdio.h>

int main (void) {
    float value1, value2;
    char operator;

    printf ("Δώσε την παράσταση.\n");
    scanf ("%f %c %f", &value1, &operator, &value2);

    if ( operator == '+' ){
        printf ("%0.2f\n", value1 + value2);
    }
    else if ( operator == '-' ){
        printf ("%0.2f\n", value1 - value2);
    }
    else if ( operator == '*' ){
        printf ("%0.2f\n", value1 * value2);
    }
    else if ( operator == '/' ){
        if ( value2 == 0 ){
            printf ("Διαίρεση δια μηδέν.\n");
        }
        else {
            printf ("%0.2f\n", value1 / value2);
        }
    }
    else
        printf ("Άγνωστη πράξη.\n");
    return 0;
}
```


Η εντολή switch

```
switch ( παράσταση ){  
    case value1:  
        program statement  
        program statement  
        ...  
        break;  
    case value2:  
        program statement  
        program statement  
        ...  
        break;  
    ...  
    case valuen:  
        program statement  
        program statement  
        ...  
        break;  
    default:  
        program statement  
        program statement  
        ...  
        break;  
}
```

```
/* Πρόγραμμα υπολογισμού απλών παραστάσεων - τιμή σύμβολο πράξης τιμή */
#include <stdio.h>

int main (void){
    float value1, value2;
    char operator;

    printf ("Δώσε την παράσταση.\n");
    scanf ("%f %c %f", &value1, &operator, &value2);
    switch (operator){
        case '+':
            printf ("%f\n", value1 + value2);
            break;
        case '-':
            printf ("%f\n", value1 - value2);
            break;
        case '*':
            printf ("%f\n", value1 * value2);
            break;
        case '/':
            if ( value2 == 0 )
                printf ("Διαίρεση δια μηδέν.\n");
            else
                printf ("%f\n", value1 / value2);
            break;
        default:
            printf ("Άγνωστη πράξη.\n");
            break;
    }
    return 0;
}
```

Μεταβλητές Boolean

//Πρόγραμμα δημιουργίας πίνακα πρώτων αριθμών

```
#include <stdio.h>
```

```
#define LIMIT 150
```

```
int main (void) {
```

```
    int p, d;
```

```
    _Bool isPrime;
```

```
    for ( p = 2; p <= LIMIT; ++p ) {
```

```
        isPrime = 1;
```

```
        for ( d = 2; d < (p/2)+1; ++d ){
```

```
            if ( p % d == 0 ){
```

```
                isPrime = 0;
```

```
            }
```

```
        }
```

```
        if ( isPrime != 0 ){
```

```
            printf ("%i ", p);
```

```
        }
```

```
    }
```

```
    printf ("\n");
```

```
    return 0;
```

```
}
```

```
#include <stdbool.h> → true == 1 , false == 0
```

