

Laborator 7 – Instrucțiuni de bază 1 (if și switch)

Instrucțiunea de tip **expresie** reprezintă o expresie corectă în C, urmată de “;”. Forma generală este:

```
expresie; /*instrucțiune expresie*/
```

Exemple

```
x=x+y; /*Instrucțiune de atribuire*/  
func(); /*Instrucțiune de apel a unei funcții*/  
a+g(); /*Instrucțiune expresie*/
```

Toate aceste exemple sunt instrucțiuni expresie. Prima instrucțiune de mai sus este o instrucțiune de atribuire. A doua instrucțiune execută apelarea unei funcții și este o instrucțiune de apel. Cea de a treia instrucțiune este o instrucțiune expresie pe care compilatorul C poate s-o evalueze corect considerând că funcția g() îndeplinește o sarcină utilă.

Instrucțiunile bloc sau instrucțiunile compuse sunt grupuri de instrucțiuni legate între ele și tratate unitar. Instrucțiunile care alcătuiesc un bloc sunt legate între ele din punct de vedere logic. Un bloc începe cu o acoladă de deschidere și se încheie cu o acoladă de închidere. Aceste instrucțiuni se utilizează în cadrul instrucțiunilor de selecție sau de iterație.

Forma generală poate fi:

Forma 1:

```
{  
instrucțiuni;  
}
```

sau

Forma 2:

```
declarații;  
instrucțiuni;  
}
```

Variabilele care sunt declarate în cadrul instrucțiunii bloc sunt definite de la execuția primei instrucțiuni până la ultima instrucțiune din bloc.

Exemplu

```
void main(void)  
{  
int y;  
...  
{  
int i;  
i=250;  
y=15*i+7;
```

```
printf(“%d\n”,i);  
printf(“%d\n”,y);  
}  
...  
}
```

Variabila i este definită de îndată ce controlul programului ajunge la prima instrucțiune din instrucțiunea bloc (i=250;). După execuția ultimei instrucțiuni printf() a instrucțiunii bloc, variabila i nu mai este definită. Variabila y a fost definită în afara instrucțiunii compuse și ea își continuă existența și după închiderea instrucțiunii compuse.

Instrucțiunea if

Este o instrucțiune de selecție care permite aplicațiilor să examineze condițiile, să specifice acțiunile ce se vor executa, în funcție de acele condiții. Sunt trei forme

Forma 1: instrucțiunea if cu o singură alternativă

```
if(expresie)  
    instructiune;
```

Forma 2: instrucțiunea if-else cu două alternative

```
if(expresie)  
    instructiune 1;  
else  
    instructiune 2;
```

Forma 3: instrucțiunea if-else cu mai multe alternative

```
if(expresie1)  
    instructiune 1;  
else  
    if(expresie 2)  
        instructiune2;  
    .  
    .  
    .  
    else  
        if(expresie n)  
            instructiune n;  
        else  
            instructiune;
```

În forma 1 a instrucțiunii if:

expresie se evaluează. Dacă expresie este evaluată ca adevărată, instrucțiunea sau blocul instrucțiune care formează ținta lui if este executată. În caz contrar se trece la execuția instrucțiunii imediat următoare instrucțiunii if.

În forma 2 a instrucțiunii if:

expresie se evaluează. Dacă expresie este evaluată ca adevărată, instrucțiunea sau blocul care formează ținta lui if instrucțiune 1 este executată. În caz contrar se va executa instrucțiunea sau blocul instrucțiune 2 care este determinat de else.

Observații:

1. Componenta else este opțională.
2. Se va executa fie codul lui if fie codul lui else dar niciodată amândouă.

În forma 3 a instrucțiunii if:

Evaluarea expresiilor are loc de sus în jos.

Există următoarele situații:

-Dacă este găsită o condiție adevărată este executată instrucțiunea asociată acesteia instrucțiune i și restul scării este omis, trecându-se la instrucțiunea imediat următoare instrucțiunii if;

-Dacă nici una dintre condiții nu este adevărată, atunci se va executa instrucțiune din ultima clauză else. Dacă această clauză nu există, atunci instrucțiunea if nu are nici un efect, neavând loc nici o acțiune.

Această structură numită instrucțiune if imbricată este foarte des întâlnită în programare și conferă aplicațiilor putere și flexibilitate.

Instrucțiunea switch

Instrucțiunea switch realizează o structură selectivă. Aspectul general al instrucțiunii switch este prezentat mai jos:

```
switch(expresie){
    case constanta 1:
        secvență de instrucțiuni 1
        break;
    case constanta 2:
        secvență de instrucțiuni 2
        break;
    case constanta 3:
        secvență de instrucțiuni 3
        break;
    ...
default
    secvență de instrucțiuni
}
```

Instrucțiunea se execută în felul următor:

Pasul 1: Se evaluează **expresie** din paranteze.

Pasul 2: Valoarea expresiei **expresie** este comparată pe rând cu fiecare din valorile constantelor exprimate în instrucțiunile case. La găsirea unei egalități, secvența de instrucțiuni asociată acelei instrucțiuni case este executată până la instrucțiunea break, după care se trece la instrucțiunea imediat următoare instrucțiunii switch. Instrucțiunea default (implicit) este executată dacă nu este găsită nici o egalitate.

Observații:

1. Instrucțiunea default este facultativă și dacă nu se află în program și nu este găsită nici o egalitate nu are loc nici o acțiune.
2. Instrucțiunea case nu poate exista de sine stătător în afara instrucțiunii switch, deși este o instrucțiune de tip etichetă.

3. Instrucțiunea switch efectuează numai teste de egalitate spre deosebire de if care poate evalua orice tip de expresie.
4. În cadrul instrucțiunii switch, două constante case nu pot avea valori identice.
5. Este posibil să nu se ajungă la instrucțiunea imediat următoare instrucțiunii switch dacă secvența de instrucțiuni selectată va defini ea însăși un alt mod de continuare a execuției programului (ex: execuția instrucțiunii de revenire dintr-o funcție, saltul la o instrucțiune etichetată etc).
6. Prezența instrucțiunii break la sfârșitul fiecărei secvențe de instrucțiuni nu este obligatorie. Dacă break lipsește după execuția secvenței de instrucțiuni alese, se va trece la execuția secvenței de instrucțiuni următoare din instrucțiunea switch.

Instrucțiunea switch este echivalentă cu următoarea instrucțiune if:

```
if(expresie==constanta1)
    secvență de instrucțiuni 1
else
    if(expresie==constanta2)
        secvență de instrucțiuni 2
    else
        if(expresie==constanta3)
            secvență de instrucțiuni 3
        else
            if
            ...
            else
                secventa de instrucțiuni
```

Exemple

1. Testați următorul program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int nota;
clrscr();
printf("Introduceti nota: ");
scanf("%d",&nota);
    if ( nota <= 4 )
    {
        printf("Nu ati invatat la programare.\n");
        printf("Ati picat examenul ! \n");
    }
    else
        printf("Ati reusit la examen ! \n");
printf("Apasati orice tasta pentru a incheia programul.");
getch();
}
```

La ce concluzie ați ajuns?

2. Să se calculeze valoarea funcției $f(x)=y$, în punctul x citit de la tastatură, unde funcția are următoarea formă:

$$f(x) = \begin{cases} x-5, & \text{daca } x < -7 \\ 12, & \text{daca } -7 \leq x < 3 \\ \frac{x^2 + x + 3}{x-5}, & \text{daca } x \geq 3 \end{cases}$$

```
#include<stdio.h>
#include<conio.h>
void main()
{
int x;
float y;
clrscr();
printf("Introduceti valoarea lui x=");
scanf("%d",&x);
if (x<-7) y=x-5;
    else
        if (x<3) y=12;
            else
                if (x==5) printf("In punctul x=%d functia nu se poate calcula",x);
                    else y=(x*x+x+3)/float(x-5);
if (x!=5) printf("Valoarea functiei f(%d) este y=%.2f",x,y);
getch();
}
```

3. Să se elaboreze un program care să simuleze un calculator de buzunar pentru operațiuni matematice simple: +,-,*,/.

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
double x,y;
char caract;
clrscr();
printf("Introduceti valoarea lui x=");
scanf("%lf",&x);
printf("Introduceti valoarea lui y=");
scanf("%lf",&y);
printf("Introduceti un operator (+ sau - sau * sau /): ");
caract=getche();
if (caract=='+')
    printf("\nRezultatul operatiei x+y=%.2lf\n",x+y);
    else if (caract=='-')
        printf("\nRezultatul operatiei x-y=%.2lf\n",x-y);
        else if (caract=='*')
            printf("\nRezultatul operatiei x*y=%.2lf\n",x*y);
            else if (caract=='/')
```

```
        if (y==0)
            printf("\nNu se poate realiza operatia (numaratorul are
valoarea 0)\n");
        else
            printf("\nRezultatul operatiei x/y=%.2lf\n",x/y);
    else printf("\nNu ati introdus unul din caracterele solicitate\n");

printf("Apasati orice tasta pentru a incheia programul");
getch();
}
```

Exercițiu propus:

4. Să se calculeze valoarea funcției $f(x)=y$, în punctul x citit de la tastatură, unde funcția are următoarea formă:

$$f(x) = \begin{cases} 2x+1, & \text{daca } x < -5 \\ x-5, & \text{daca } -5 \leq x < -2 \\ \frac{x+1}{x(x-3)}, & \text{daca } -2 \leq x < 8 \\ x+6, & \text{daca } x \geq 8 \end{cases}$$

5. Să se scrie un program care să permită utilizatorului să selecteze o culoare astfel: tasta 1 selectează culoarea roșie, tasta 2 culoarea verde, tasta 3 culoarea albastră.

```
#include <stdio.h>
#include <conio.h>
void main() {
int culoare = 0;
clrscr();
printf("Introduceti o culoare (1: rosu,2: verde,3: albastru):\n");
scanf("%d", &culoare);
switch (culoare) {
    case 1:
        printf("ati ales culoarea rosie\n");
        break;
    case 2:
        printf("ati ales culoarea verde\n");
        break;
    case 3:
        printf("ati ales culoarea albastru\n");
        break;
    default:
        printf("nu ati ales nicio culoare\n");
}
printf("Apasati orice tasta pentru a incheia programul");
getch();
}
```

6. Să se realizeze un program care să simuleze operații bancare efectuate asupra unui cont existent. Programul trebuie să permită introducerea sumei inițiale din cont, alegerea unei operațiuni (restituire/depunere/anulare) și să efectueze operațiile necesare asupra sumei existente.

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
double suma,suma1;
char operatie;
clrscr();
printf("Suma existenta in cont:");
scanf("%lf",&suma);
printf("R-Restituire\nD-Depunere\nS- Stop\n");
printf("Selectati optiunea dorita:\n");
scanf("\n%c",&operatie);
switch(operatie)
{
case 'r':
case 'R':
printf("Ce suma doriti?");
scanf("%lf",&suma1);
if(suma1>suma)
{
printf("Nu se poate onora cererea.\n");
break;
}
suma-=suma1;
break;
case 'd':
case 'D':
printf("Ce suma doriti sa depuneti?");
scanf("%lf",&suma1);
suma+=suma1;
break;
case 's':
case 'S':
break;
}
printf("Suma existenta in cont este %lf\n",suma);
printf("Apasati orice tasta pentru a incheia programul");
getch();
}
```

6. Să se scrie un program prin care, la apăsarea unui taste numerice să se precizeze tipul cifrei selectate: 0 pentru zero, pătrat perfect, număr par sau număr prim. Dacă se selectează orice altceva, să se apară un mesaj de atenționare.

Secvența switch este:

```
switch (n) {
  case 0:
    printf("Ati tipărit 0.");
    break;
  case 1:
  case 4:
  case 9:

    printf("Ati ales o cifră pătrat perfect.");
    break;
  case 2:
    printf("Ati ales o cifră pară.");
  case 3:
  case 5:
  case 7:
    printf("Ati ales o cifră număr prim.");
    break;
  case 6:
  case 8:
    printf("Ati ales o cifră pară.");
    break;
  default:
    printf("Nu ati ales o cifră.");
    break;
}
```

Să se introducă această secvență într-un program C funcțional.

7. Modificați programul de la exemplul 3 astfel încât să fie realizat folosind instrucțiunea switch.