

5. INSTRUCȚIUNI DE DECIZIE

În funcție de context, la un moment dat, un program de calcul trebuie să aibă posibilitatea să selecteze un anumit set de instrucțiuni pe care să le execute. Cu alte cuvinte, el trebuie astfel realizat încât să fie capabil să ia o anumită decizie. Pentru programarea deciziilor, limbajul C pune la dispoziția programatorului trei instrucțiuni: **if**, **if...else** și **switch**.

5.1. Instrucțiunea **if**

Instrucțiunea **if** este instrucțiunea de bază în luarea deciziilor în limbajul C. Ca structură, aceasta este similară cu bucla **while**. Considerăm pentru exemplificare următorul program:

Exemplul 5.1.

```
/* Programul ex_5_1 */
void main(void)
{
    int n_car=0,    n_cifre=0;
    char car;
    clrscr(); printf("Introduceti un sir de caractere \n");
    while ((car=getche()) != 13)
    {
        n_car++;
        if(car>=48 && car<=57) n_cifre++;
    }
    printf("\nNumarul de caractere introdus este %2d dintre care %2d sunt cifre",n_car,n_cifre);    getch ();
}
```

Acest program solicită introducerea unui șir de caractere, determină numărul caracterelor introduse, precum și câte dintre acestea sunt cifre. Pentru contorizarea numărului de caractere și a numărului de cifre se folosesc variabilele de tipul întreg n_car și respectiv n_cifre . Șirul de caractere este citit cu ajutorul funcției **getche** în cadrul unei bucle **while**, care se termină în momentul în care s-a apăsat tasta <Enter>. În cazul în care codul zecimal al caracterului citit și atribuit variabilei **car** este diferit de 13 (codul tastei <Enter>), se incrementează variabila n_car după care se testează dacă respectivul caracter este o cifră, folosind instrucțiunea **if** ($car \geq 48 \ \&\& \ car \leq 57$). Pentru aceasta se evaluează expresia logică $car \geq 48 \ \&\& \ car \leq 57$. Dacă expresia este adevărată, se execută instrucțiunea de incrementare a numărului de cifre $n_cifre++$. În caz contrar, această instrucțiune este ignorată și se revine la începutul buclei pentru citirea unui nou caracter.

Din cele prezentate, concluzionăm că structura instrucțiunii **if** cuprinde:

- o instrucțiune de forma: *if (expresie logică)* care are rolul de a decide dacă instrucțiunea, sau setul de instrucțiuni, care urmează se va executa. Această instrucțiune, similară cu cea care deschide bucla **while**, începe cu cuvântul cheie **if** și nu se termină cu caracterul ";".
- corpul instrucțiunii de decizie, format dintr-o instrucțiune sau un set de instrucțiuni.

Pentru exemplificare, considerăm programul următor în care se rezolvă ecuația de gradul I, $ax+b=0$.

Exemplul 5.2.

```
/* Programul ex_5_2 */
void main(void)
{
    float a, b, x;
    clrscr();
    printf("\nIntroduceti coeficientii a si b >"); scanf("%f %f",&a,&b);
    if (a != 0)
    {
        x = -b/a; printf("\nRadacina ecuatiei %6.2f+x+%6.2f=0 este x=%f",a,b,x);
    }
    getch ();
}
```

Programul solicită introducerea coeficienților polinomiali a și b , iar dacă a este diferit de 0, atunci se calculează rădăcina $x = -b/a$ și se afișează rezultatul. În acest caz, corpul instrucțiunii de decizie se compune din două instrucțiuni simple, motiv pentru care a fost necesară utilizarea acoladelor.

De menționat că este posibilă înlănțuirea mai multor instrucțiuni **if**, adică este permisă utilizarea unei instrucțiuni **if** în corpul altei instrucțiuni **if**.

5.2. Instrucțiunea **if...else**

Structura generală a instrucțiunii **if .. else** este:

```
if (expresie logică)
    setul de instrucțiuni 1
else
    setul de instrucțiuni 2.
```

Cu alte cuvinte, se completează structura instrucțiunii **if**, după setul de instrucțiuni ce alcătuiesc corpul acesteia, cu o instrucțiune formată din cuvântul cheie **else** (fără caracterul ";"), urmată de instrucțiunea, sau setul de instrucțiuni, ce se va executa în cazul în care expresia logică este falsă.

Ca prim exemplu de utilizare a instrucțiunii de decizie **if .. else** se prezintă programul din exemplul 5.3, care este o completare a programului **ex_5_2**, astfel încât să afișeze mesajul: "Ecuație degenerată", dacă $a=0$.

Exemplul 5.3.

```
/* Programul ex_5_3 */
void main(void)
{
    float a,b,x;
    clrscr ();
    printf ("\nIntroduceti coeficientii a si b >"); scanf ("%f %f",&a,&b);
    if (a != 0)
    {
        x=-b/a; printf ("\nRadacina ecuatiei %6.2f*x+%6.2f=0 este x=%f",a,b,x);
    }
    else
        printf ("\nzcuatie degenerata");
    getch ();
}
```

Dacă lansăm în execuție acest program și introducem pentru a și b valorile 5 și 2.5, separate între ele printr-un spațiu (conform descriptorului de format al funcției **scanf**), atunci programul va afișa pe ecran mesajul: “*Rădăcina ecuației $5x+2.5=0$ este $x=-0.5$* ”, ca rezultat al execuției instrucțiunilor din cadrul instrucțiunii **if**. Dacă în schimb vom introduce valorile 0 și 2.5, atunci programul va afișa mesajul: “*Ecuație degenerată*” care reprezintă rezultatul execuției apelului funcției **printf** din corpul instrucțiunii **else**.

Deoarece, ca și în cazul buclelor, limbajul de programare C tratează ansamblul instrucțiunilor de decizie ca instrucțiuni compuse, este posibilă utilizarea acestora în cadrul corpului buclelor sau invers, utilizarea buclelor ca instrucțiuni în corpul instrucțiunilor de decizie. Pentru exemplificare, se prezintă programul din exemplul 5.4, care desenează pe ecran o tablă de șah.

Exemplul 5.4.

```
/* Programul ex_5_4 */
void main (void)
{
    int i,j;
    clrscr ();
    for(i=1; i<=8; i++)
    {
        gotoxy(32,7+i);
        for (j=1; j<=8; j++)
        {
            if ((i+j)%2 == 0) printf ("\xDB\xDB");
            else printf ("");
        }
        getch ();
    }
}
```

Algoritmul se bazează pe faptul că un pătrat este alb, dacă suma dintre linia și coloana în care se află este pară și negru dacă această sumă este impară.

Pentru implementarea algoritmului se utilizează două bucle **for** și o instrucțiune **if..else**. Prima buclă, avînd ca variabilă de control pe i , contorizează numărul liniilor tablei de șah, iar a doua, avînd ca variabilă de control pe j , contorizează numărul coloanelor dintr-o astfel de linie. Pe ecran tabla de șah va fi generată începînd din coloana 32 și linia 8. Astfel, pentru o anumită linie i a tablei de șah se evaluează mai întîi expresia $7+i$, după care, prin apelul funcției **gotoxy** se realizează plasarea acesteia pe ecran în poziția de început. În continuare, se parcurge bucla interioară, pentru generarea celor opt pătrate ce alcătuiesc linia curentă a tablei de șah. În cadrul acestei bucle se utilizează instrucțiunea **if..else** pentru a decide dacă pătratul generat este alb sau negru. În acest sens, se utilizează expresia logică $(i+j)\%2==0$. Dacă acest rest este 0 ($i+j$ este un număr par), se va executa corpul instrucțiunii **if** care constă din apelul funcției **printf** pentru tipărirea de două ori a caracterului grafic definit de secvența escape \backslash DB. Tipărirea succesivă a două caractere este necesară pentru generarea pătratului, deoarece înălțimea unui caracter este mai mare de aproximativ două ori decît lățimea sa. În cazul în care restul împărțirii este 1 ($i+j$ este un număr impar), se va executa corpul instrucțiunii **else**, care constă din apelul funcției **printf**, pentru a lăsa două spații, adică pentru a delimita pătratul negru. Acest proces este repetat pînă cînd vor fi generate toate cele 64 de pătrate ale tablei de șah.

În cazul înlănțuirii instrucțiunilor de decizie **if** și **if..else** se precizează faptul că, în absența acoladelor, o instrucțiune **else** este asociată ultimei instrucțiuni **if** care nu are asociată o instrucțiune **else**. Pentru a înțelege această regulă, să presupunem că dorim să scriem un program de calcul care să caracterizeze o zi de vară, după valoarea temperaturii pe care o introducem de la tastatură. Astfel, dacă temperatura este cuprinsă între 20°C și 30°C programul va afișa mesajul: “*Este o zi plăcută*”, iar dacă temperatura este mai mare de 30°C, mesajul: “*Este o zi caniculară*”. Dacă vom scrie programul ca în exemplul 5.5 și vom tasta pentru temperatură valoarea 15, acesta va afișa pe ecran mesajul eronat: “*Este o zi caniculară*”.

Exemplul 5.5.

```
/* Programul ex_5_5 */
void main(void)
{
    float temp;
    clrscr ();
    printf ("\nIntroduceti valoarea temperaturii > "); scanf ("%f",&temp);
    if(temp < 30)
    if(temp >=20)
        printf ("\nEste o zi placuta");
}
```

```

else
    printf("\nEste o zi caniculara");
getch ();
}

```

Eroarea s-a obținut deoarece instrucțiunea **else** este asociată celei de-a doua instrucțiuni **if**, formînd împreună o structură **if..else**, care la rîndul ei constituie corpul primei instrucțiuni **if**.

Astfel, la tastarea valorii 25 programul funcționează corect. Deoarece expresia logică de test a primei instrucțiuni **if** este adevărată, se trece la execuția corpului acesteia format din instrucțiunea compusă **if..else**. Cum și expresia logică de test a celei de-a doua instrucțiuni este adevărată, programul va afișa mesajul corect "*Este o zi plăcută*". Dacă în schimb se tastează valoarea 15, expresia logică a instrucțiunii **if..else** este falsă, deci se va executa corpul instrucțiunii **else** care afișează mesajul eronat "*Este o zi caniculara*". Pentru a corecta programul este necesară delimitarea corpului primei instrucțiuni **if** cu ajutorul acoladelor, ca în programul din exemplul 5.5a. Aceste acolade vor masca a doua instrucțiune **if**, astfel ca instrucțiunea **else** este asociată cu prima instrucțiune **if** și programul va funcționa corect.

Exemplul 5.5.a

```

/* Programul ex_5_5_a */
void main (void)
{
    float temp;
    clrscr ();
    printf("\nIntroduceti temperatura >");    scanf("%f",&temp);
    if (temp < 30)
    {
        if(temp >=20)
            printf ("\nEste o zi placuta");
        }
    else
        printf("\nEste o zi caniculara");
    getch ();
}

```

În această variantă programul nu va afișa nici un mesaj în cazul în care temperatura este mai mică de 20°C. Dacă dorim completarea programului astfel încît pentru temperaturi mai mici de 20°C să se afișeze mesajul "*Este o zi răcoroasă*", atunci acesta se va scrie,utilizînd două instrucțiuni **if..else** înlănțuite, ca în exemplul 5.5.b.

Exemplul 5.5.b

```

/* Programul ex_5_5b */
void main (void)
{
    float temp;
    clrscr ();
    printf("\nIntroduceti temperatura > ");    scanf ("%f", &temp);
    if(temp < 30)
        if (temp > 20)
            printf ("\nEste o zi placuta");
        else
            printf ("\nzste o zi racoroasa");
    else
        printf("\nEste o zi caniculara");
    getch ();
}

```

Pentru a preîntîmpina situații de tipul celei prezentate în exemplul analizat, se recomandă ca atunci cînd există ambiguități privind înlănțuirea instrucțiunilor de decizie sau chiar a buclelor, să se utilizeze acoladele pentru delimitarea strictă a corpurilor acestora.

În programul din exemplul 5.6 se rezolvă ecuația de gradul II, $ax^2+bx+c = 0$.

Exemplul 5.6.

```

/* Programul ex_5_6 */
#include <math.h>
void main (void)
{
    float a,b,c,x1,x2,x1r,x2r,delta;
    clrscr ();
    while (1)
    {
        printf ("\nApasati Enter pentru a continua si Esc pentru a termina programul ")
        if (getch( )==27) break;
        clrscr ();
        printf ("\nIntroduceti valorile coeficientior a b c >");    scanf("%f %f %f",fa,&b,&c);
        if (a != 0)
        {
            /* Ecuatie de gradul II */
            delta=b*b-4*a*c;

```



```

case '+': printf("%8.2f+%8.2f=%8.2f",a,b,a+b);    break;
case '-':      printf("%8.2f-%8.2f=%8.2f",a,b,a-b);    break;
case '*': printf("%8.2f*%8.2f=%8.2f",a,b,a*b);    break;
case '/':      printf("%8.2f/%8.2f=%8.2f",a,b,a/b);    break;
default:      printf(" Operatie necunoscuta ");

```

Pentru efectuarea de operații repetate, programul utilizează o buclă infinită care poate fi părăsită prin apăsarea tastei <Esc>. Dacă se decide continuarea execuției, prin apăsarea tastei <Enter>, programul solicită introducerea expresiei sub forma a două numere, separate între ele prin semnul operației aritmetice ce urmează a fi efectuată cu acestea. Selecția se face în corpul instrucțiunii **switch**, folosind semnul operației citit în variabila *oper* de tipul caracter, cu ajutorul cuvântului cheie **case**.

Astfel, dacă valoarea lui *oper* este egală cu una din constantele caracter "+", "-", "*" sau "/" ce urmează după cuvintele **case**, este selectat cazul respectiv și se execută setul de instrucțiuni care urmează. La sfârșitul fiecărui caz se folosește instrucțiunea **break** pentru a părăsi corpul instrucțiunii de selecție.

Dacă din greșeală, pentru semnul operației s-a tastat un caracter diferit de cele patru caractere recunoscute, nu este selectat nici un caz și se execută instrucțiunea ce urmează cuvântului cheie **default**, adică se afișează mesajul: "Operatie necunoscuta".

Din cele prezentate concluzionăm că structura generală a acestei instrucțiuni cuprinde:

- o instrucțiune de forma *switch (expresie)*, compusă din cuvântul cheie **switch** urmat de o pereche de paranteze rotunde în interiorul cărora poate fi o variabilă sau o expresie de tipul întreg sau caracter. Nu este permisă utilizarea variabilelor sau a expresiilor de tipul real. Această instrucțiune nu se termină cu caracterul punct și virgulă (";").

- o pereche de acolade, care delimitează corpul instrucțiunii de decizie, format din instrucțiunile ce alcătuiesc cazurile posibile. Definierea unui caz se realizează prin utilizarea cuvântului cheie **case** urmat de o constantă de tipul caracter sau întreg și caracterul ":". Sfârșitul unui caz este marcat de instrucțiunea **break**, care are ca efect părăsirea instrucțiunii de selecție. Instrucțiunile cuprinse între cuvintele cheie **case** și **break** formează corpul de instrucțiuni al cazului respectiv și nu este necesar a fi delimitat prin acolade.

Este posibilă definierea unui caz implicit cu ajutorul cuvântului cheie **default** urmat de caracterul ":". Acesta delimitează setul de instrucțiuni ce se va executa în situația în care nu s-a selectat nici un caz datorită faptului că valoarea expresiei din parantezele instrucțiunii **switch** nu este egală cu nici una din constantele instrucțiunilor **case**. Definierea acestui caz este opțională și prin urmare el poate lipsi.

În încheiere, se precizează faptul că instrucțiunea **break** joacă un rol esențial în funcționarea instrucțiunii de selecție multiplă **switch**. Dacă aceasta lipsește, după selectarea unui caz și execuția instrucțiunilor din corpul său, se vor executa și instrucțiunile din corpul cazurilor cel urmează. Există situații în care nu se asociază fiecărei instrucțiuni **case** o instrucțiune **break**. Pentru exemplificare, să considerăm că dorim modificarea programului anterior astfel încât acesta să execute operația de înmulțire și în cazul în care se tastează caracterul "x". În acest sens setul de instrucțiuni:

```

case '*': printf("\n = %f",a*b);    break;

```

se înlocuiește cu setul de instrucțiuni:

```

case '*':
    case 'x': printf("\n = %f",a*b);    break;

```

Deoarece după primul **case** nu s-a folosit instrucțiunea **break** la tastarea caracterului "*", se va continua execuția cu instrucțiunea aferentă cazului *case "x"*, adică se va afișa rezultatul înmulțirii după care instrucțiunea de selecție este părăsită.