UNIVERSITATEA TEHNICĂ A MOLDOVEI

PROGRAMAREA CALCULATOARELOR

Indicaţii de laborator

Lucrarea de laborator nr.2
Programe ramificate

1. Scopul

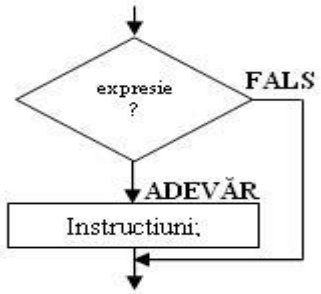
Studierea, utilizarea şi obţinerea deprinderilor practice de elaborare şi depanare a programelor ramificate (if-else, switch-break).

2. Descrierea temei

2.1. Instrucţiunea *if* şi *if-else*

Instrucţiunea ***if*** permite programarea unei structuri de decizie în care o condiţie (rezultatul evaluă rii unei expresii) determină executarea sau neexecutarea secvenţei de instrucţiuni.

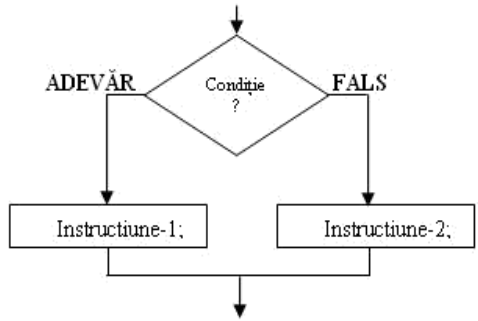
Sintaxa instrucţiunii ***if***	Schema bloc

if (expresie)
{
instrucţiuni;
}
	

Modul de funcţionare: la început se calculează valoarea expresiei din parantezele rotunde ale instrucţiuniei **if**. Dacă valoarea acesteia este **ADEVĂR**, atunci se vor executa toate instrucţiunile cuprinse de parantezele ondulate("**{ }**"), pe care ulterior le vom numi corp al funcţiei. În caz contrar se ignoră instrucţiunile din corpul lui **if** şi se trece la următoarele rânduri de cod.

Instrucţiunia ***if-else*** permite programarea unei structuri de decizie în care o condiţie determină executarea unei secvenţe de program din două alternative.

Sintaxa instrucţiunii ***if-else***	Schema bloc

if (condiţie)
{
Instrucţiune-1;
}
else
{
Instrucţiune-2;
}
	

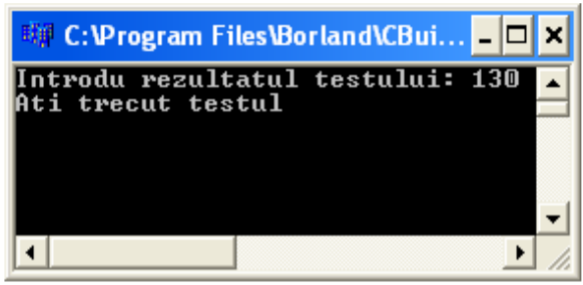
Modul de funcţionare: În primul rînd se calculează valoarea expresiei din parantezele rotunde ale instrucţiunii ***if,*** dacă valoarea acesteia este **ADEVĂR** , atunci se vor executa toate instrucţiunile din corpul lui ***if*** . În caz contrar se ignoră instrucţiunile din corpul lui ***if*** şi se execută cele din corpul lui ***else***.

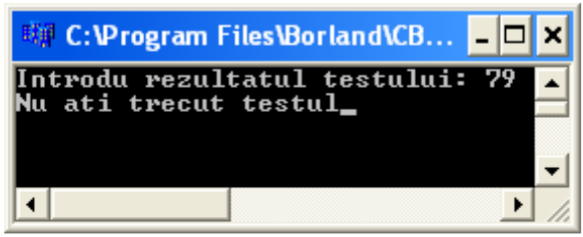
Exemplu: Să se scrie un program ce va utiliza instrucţiunile ***if-else***:

```
#include<conio.h>
#include<stdio.h>
void main()
{ int n;
  printf("Introdu rezultatul testului:");
  scanf("%d",&n);
  if(n>=100)
   {
    printf("Ati trecut testul");
   }
  else
   {
    printf("Nu ati trecut testul");
```

```
    }
    getch();
    }
```

Rezultatul execuţiei:





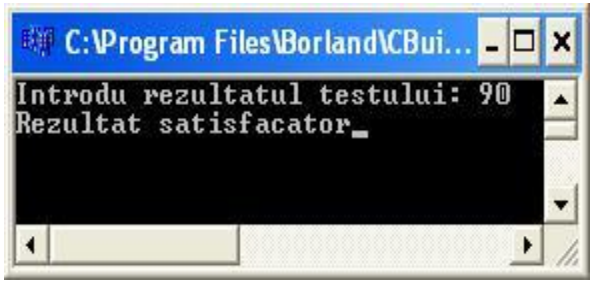
În practica programării, din când în când apare necesitatea alegerii unei operaţiuni din mai multe posibile, pentru aceasta se permite utilizarea lanţ ului de instrucţiuni ***if-else-if*** sau instrucţiunea ***switch(),*** pe care o vom examina ceva mai târziu.

Exemplu: Să se scrie un program ce va utiliza lanţul de instrucţiuni ***if-else-if***:

```
#include<conio.h>
#include<stdio.h>
void main()
{
  int n;
  printf("Introdu rezultatul testului: ");
  scanf("%d",&n);
  if(n>=100)
```

```
      {
       printf("Rezultat maxim");
      }
  else
      {
       if(n>=90)
          {
           printf("Rezultat satisfacator");
          }
       else
          {
            printf("Ati picat testul");
          }
       }
      getch();
}
```

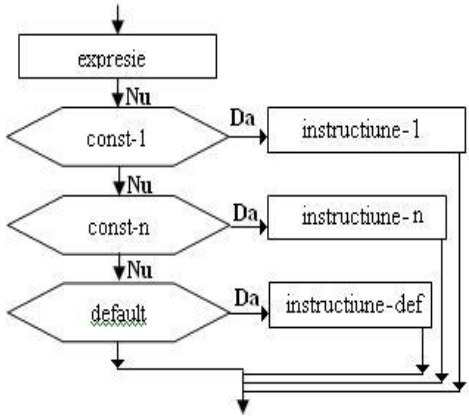
Rezultatul execuţiei:



2.2. Instrucţiunea *SWITCH*

Instrucţiunea ***switch*** permite executarea unei singure secvenţe din mai multe alternative, în funcţie de valoarea unei expresii. Sintaxa instrucţiunii ***switch*** este următoarea:

Sintaxa instrucţiunii ***switch***	Schema bloc

switch (expresie)
{
case const-1:
{
instructiune-1; **break;**
}
…
case const-n:
{
instructiune-n; **break**;
}
default:
{
instructiune-def;
}
}
	

Modul de funcţionare: Întâi se calculează valoarea expresiei „***expresie***”. Dacă aceasta este egală cu una din constantele corespunzătoare fiecărei etichete ***case***, se execută instrucţiunea din corpul constantei respective. Fiecare grup ***case*** trebuie terminat cu o instrucţiune ***break***. În cazul în care aceasta lipseşte, se execută toate instrucţiunile, începând cu constanta selectată până la ultimul ***case***. Când este întâlnită o instrucţiune ***break*** în construcţia ***switch***, programul execută un salt la linia de cod ce urmează după corpul lui ***switch***. Instrucţiunea corespunzătoare cazului implicit (***default***) este executată când expresia nu ia nici una din valorile constantelor. Default este

opţional ş i, dacă nu este prezent, nu are loc nici o acţiune dacă nu se găseşte o constantă potrivită.

Pentru înţelegerea completă trebuie menţionate următoarele:

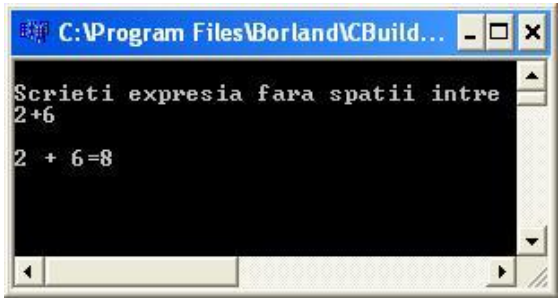
a) ***switch*** diferă de ***if*** prin aceea că testează doar egalitatea, în timp ce ***if*** poate să evalueze orice tip de expresie relaţională sau logică;
b) în acelaşi ***switch*** nu pot exista două constante ***case*** cu valori identice. Desigur, două instrucţiuni ***switch***, una inclusă în cealaltă, pot să aibă aceeaşi constantă ***case***.
c) Dacă în instrucţiunea ***switch*** sunt utilizate constante de tip caracter, ele sunt automat convertite în întregi.

Exemplu: Să se scrie un program ce va efectua operaţii cu numere întregi de forma OPERAND1 operator OPERAND2 (2*4):

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
int op1,op2,rez;
char op;
printf("\nScrieti expresia fara spatii intre
operanzi si operator\n");
scanf("%d%c%d",&op1,&op,&op2);
switch(op)
{
 case '+':
    {
     rez=op1+op2; break;
    }
 case '-':
    {
     rez=op1-op2; break;
    }
 case '*':
```

```
    {
     rez=op1*op2; break;
    }
 case '/':
    {
     if (op2!=0)
      {
       rez=op1/op2;
      }
     else
      {
       printf("Nu se imparte la zero!!!!");
       getch(); exit(0);
      }
     break;
    }
 default:
    {
      printf("Nu a avut loc o operatie!");
      getch();
      exit(1);
    }
}
printf("\n%d %c %d=%d", op1, op,op2, rez);
getch();
}
```

Rezultatul execuţiei:



3. Variantele propuse spre elaborare

Să se scrie un program care va calcula valoarea funcţiei în funcţie de condiţii. În program să se utilizeze instrucţiunile ***if***, ***if-else***, ***if-else-if*** şi ***switch***:

Nr.	Funcţia	Condiţia
1	$M=\begin{cases}\pi b^3+\lvert a+x\rvert, 1\le x\le 3\\ e^{ax}+\sqrt[3]{xb}, x<1\\ 4tga, x>3\end{cases}$	1≤x≤3 x<1 x>3
2	$A=\begin{cases}\pi\sqrt{x^2+b}-b^2\\ \sin^3(x+a)+\sqrt[3]{x^2+b}\\ bx\cos(x)+a\end{cases}$	x<1.5 x=1.5 x>1.5
3	$L=\begin{cases}\lvert x^{y/x}-\sqrt[3]{y/x}\rvert\\ \cos^2 x^3-e^{x/a}\\ \pi e^{-ax}\sqrt{a^2+1.5}\end{cases}$	x<1.2 x=1.2 x>1.2
4	$D=\begin{cases}\pi\lvert a+x\rvert\\ (bx^2-a)/(e^{ax}-1)\\ x^2a/4\end{cases}$	x>a x=a x<a
5	$Z=\begin{cases}e^{2y}\ln(a+x+y^2)\\ \sqrt[3]{x+y}\\ \cos^2 x+\sin y^4+a^2\end{cases}$	-1≤x*y≤6 x*y<-1 x*y>6

6	$C = \begin{cases} 1.8\cos^2 h + a \\ 1.8\pi ah \\ \cosh + a\sin^2 h \end{cases}$	h>a h=a h<a
7	$V = \begin{cases} \pi d^3 - \sqrt{\lvert d - g \rvert} \\ dg - e^{-d} \\ \sin dg + 1 \end{cases}$	1≤d*2≤9 d*2<1 d*2>9
8	$B = \begin{cases} \pi e^{-\sqrt{w}} \cos(bx / w) \\ \sqrt{wx \sin 2w + e^{-2x}(w + b)} \\ tg^2(w + x + b) \end{cases}$	w<0.2 w=0.2 w>0.2
9	$L = \begin{cases} 2\cos(x - \pi / 6) \\ x^2 / a + \cos(x + b)^3 \\ \lvert x / 2a \rvert + \sin^2(x + 1) \end{cases}$	x<1.2 1.2≤ x≤3.9 x>3.9
10	$K = \begin{cases} 1.5\cos^2 v + u \\ u^2 / 3 + v^2 + \pi \\ \lvert 3tgv + u^3 \rvert \end{cases}$	-1≤v*u≤5 v*u<-1 v*u>5
11	$J = \begin{cases} (a + b) / (e^x + \cos x) \\ (a + b) / (x + 1) \\ e^x + \sin x + \pi \end{cases}$	x<2.8 2.8≤ x≤6 x>6
12	$G = \begin{cases} ca / i + bi^2 \\ i + 2i + 2 \\ ai + bi^3 \end{cases}$	i<4 4≤i≤6 i>6

13	$F = \begin{cases} a \ln x + \sqrt[3]{\lvert a - x \rvert} \\ 2a \cos x + 3x^2 \end{cases}$	x>1 x≤1
14	$T = \begin{cases} \sqrt{at^2} + b \sin t + a \\ at + b + \pi \\ \lvert 3ta^2 + \cos^3 t + 1 \rvert \end{cases}$	t<0.1 t=0.1 t>0.1
15	$K = \begin{cases} a \sin(i^2 + 1) \\ \cos(i + 1/n) \\ \lvert tgn + n^3 \rvert + \pi \end{cases}$	i<1.5 i=1.5 i>1.5
16	$WD = \begin{cases} \sin(2/k) \\ 1/k \\ \lvert k^2 \rvert + \pi \end{cases}$	k<1 k=1 k>1

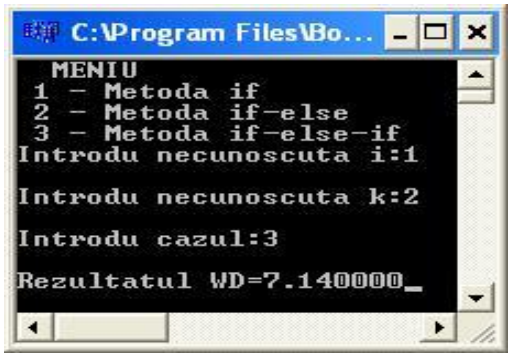
Exemplu: Să se scrie un program care va calcula valoarea funcţiei *WD* în funcţie de condiţii (vezi varianta 16). În program să se utilizeze instrucţiunile ***if***, ***if-else***, ***if-else-if*** şi ***switch***:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
# define Pi 3.14
# main()
{ int k,key; float WD; printf(" MENIU
  "); printf("\n 1 - Metoda if ");
  printf("\n 2 - Metoda if-else ");
  printf("\n 3 - Metoda if-else-if
  "); printf("\nIntrodu necunoscuta
  k:"); scanf("%d",&k);
  printf("\nIntrodu cazul:");
  scanf("%d",&key);
```

```
 switch(key)
 {
  case 1:
   {
    if(k<1)
     {
    WD=sin(2/k);}
    if(k==1)
     { WD=1/k;}
  if(k>1)
   { WD=abs(pow(k,2))+Pi;}
   printf("Rezultatul WD=%f",WD);
  break;
 }
case 2:
 {
  if(k<1)
   { WD=sin(2/k);}
  else
   {printf("\nPentru k<1,nu sunt sol.");}

  if(k==1)
  { WD=1/k;}
  else
  {printf("\nPentru k=1,nu sunt sol.");}

  if(k>1)
  { WD=abs(pow(k,2))+Pi;}
  else
  {printf("\nPentru k>1,nu sunt sol.");}

  printf("\nRezultatul WD=%f",WD);
  break;
 }
  case 3:
   {
    if(k<1)
```

```
    { WD=sin(2/k);}
    else
    {
     if(k==1)
      {WD=1/k;}
     else
      {WD=abs(pow(k,2))+Pi;}
      }
      printf("\nRezultatul WD=%f",WD);
      break;
     }
    default:{printf("\nNu este asa caz!");}
}//end switch
getch();}
```

Rezultatul execuţiei:



4. Întrebări pentru verificarea cunoştinţelor

1. Prezentaţi sintaxa instrucţiunii *if*.
2. Prezentaţi sintaxa instrucţiunii *if else*.
3. Prezentaţi sintaxa instrucţiunii *switch.*
4. Explicaţi modul de funcţionare a instrucţiunii *if*.
5. Explicaţi modul de funcţionare a instrucţiunii *if else*.
6. Explicaţi modul de funcţionare a instrucţiunii *switch*.
7. Explicaţi instrucţiunea *break.*
8. Prezentaţi schema bloc a instrucţiunii *if*.
9. Prezentaţi schema bloc a instrucţiunii *if else*.
10. Prezentaţi schema bloc a instrucţiunii *switch.*