1. **INSTRUCŢIUNILE IF ŞI SWITCH**
   1. **SCOPUL LUCRĂRII**

Lucrarea are ca scop familiarizarea studentului cu instrucţiunea condiţională (de decizie, de ramificaţie), în formele ***IF-THEH*** şi ***IF-THEN-ELSE***.

Se exersează şi versiunea ***SWITCH***.

**2.2 BREVIAR TEORETIC**

**Instrucţiunile condiţionale** determină programele să testeze diferite condiţii şi, în funcţie de acestea, să decidă execuţia anumitor comenzi.

Avem la dispoziţie **instrucţiunile condiţionale**:

* **if( )** - execută comenzile dorite atunci când o condiţie (scrisă între paranteze) este adevarată.
* **if( ) ... else** - execută anumite comenzi când o conditie (scrisă între paranteze) este adevarată şi alte comenzi când această condiţie este falsă.
* **switch** - selectează care comandă va fi executată.

**A. Instructiunea "if"**

"**if( )**" (*dacă*) este cea mai simplă instrucţiune condiţională.

Forma generală a acestei instrucţiuni este:

* **if (*condiţie*) {  
      // Codul care va fi executat dacă este Adevarată condiţia  
  }**

unde '***condiţie***' poate fi orice expresie alcatuită cu operatori raţionali, de egalitate şi logici.

Dacă rezultatul evaluării condiţiei este TRUE, se execută codul dintre acolade, în caz contrar, când condiţia returnează FALSE, se trece peste acest cod.

**B. Instrucţiunea "if ... else"**

Folosind instrucţiunea "**if() ... else**" (*dacă ... altfel*), putem stabili comenzi care să fie executate şi când condiţia instrucţiunii "if( )" este FALSE.

Forma generala a instructiuni "if() ... else" este:

**if (*condiţie*) {  
    // codul care va fi executat dacă este Adevarată condiţia  
}  
else {  
    // codul ce va fi executat daca condiţia este falsă  
}**

unde 'condiţie' poate fi orice expresie logică.

Dacă rezultatul condiţiei este TRUE, se execută codul dintre primele acolade, care aparţin de "if()", în caz contrar, când condiţia returnează FALSE, sunt executate comenzile din acoladele de la "else".

**Formula "else if( )"**

Cu "if() ... else" sunt posibile execuţiile a doar două opţiuni, cea de la "if( )" sau de la "else". Dar sunt situaţii în care avem mai multe opţiuni, caz în care se foloseşte formula "**else if( )**" (*altfel dacă*).

Cu aceasta se pot crea şi alte opţiuni (suplimentare) între cele două

Sintaxa generală este:

**if (*condiţie 1*) {  
    // codul care va fi executat dacă este Adevarată condiţia 1  
}  
else if (*condiţie 2*) {  
    // codul ce va fi executat daca prima condiţie este Falsă şi este Adevarată condiţia 2  
}  
else if (*condiţie 3*) {  
    // codul care va fi executat daca primele două condiţii sunt False şi este Adevarată condiţia 3  
}  
// ...   
else {  
    // codul executat dacă toate condiţiile sunt False  
}**

Pot fi adaugate oricate optiuni "else if".

**B. Instrucţiunea switch**

Această instrucţiune e folosită pentru a compara o valoare cu altele dintr-o listă şi, în funcţie de acea valoare, se execută codul asociat ei în lista "**switch**".

Sintaxa generala a instructiuni "switch" este:

**switch (expresie) {  
case valoare1:  
    cod executat dacă expresie = valoare1  
    break;  
case valoare2:  
    cod executat dacă expresie = valoare2  
    break;  
case valoare3:  
    cod executat pt. expresie = valoare3  
    break;  
default :  
    cod executat dacă expresie e diferit de valoare1, valoare2 sau valoare3  
}**

Prima dată este evaluată expresia scrisă între paranteze rotunde, la "switch( )", apoi valoarea expresiei este comparată pe rând cu fiecare valoare determinată de "case".

Dacă se găseşte o identitate, se execută codul asociat acelui "case", apoi se iese din instrucţiunea "switch".

Dacă, parcurgand fiecare "case", nu se găseşte o egalitate, se execută codul de la "default".

Prin folosirea lui "break" se opreşte parcurgerea corpului instrucţiunii atunci când s-a găsit o valoare egală cu 'expresie' şi se iese din "switch".

Instructiunea "switch" poate inlocui un şir de condiţii cu "else if".

**C. Operatori logici şi instrucţiunile condiţionale**

Operatorii logici sunt "**&&**" si "**||**".

Rolul lor este asemănător cu cel al operatorilor relaţionali şi de egalitate, în sensul că şi aceştia compară valoarea a doi operanzi.

Operatorii logici compară operatii, expresii de cod, iar rezultatul lor este unul din valorile TRUE (*Adevărat*) sau FALSE (*Fals*).

* **exp1 && exp2**   - Returnează True dacă şi "exp1" şi "exp2" sunt Adevărate; altfel, returneaza False.
* **exp1 || exp2**   - Returneaza True dacă oricare dintre "exp1" sau "exp2" e Adevarat; altfel, dacă amândouă sunt false, returnează False.
* **exp1 && exp2 || exp3**   - Returnează True dacă expresia (exp1 && exp2) sau "exp2" e Adevarată; altfel, returnează False.

Datorita rezultatului True sau False pe care-l dau aceşti operatori logici, ei sunt folosiţi în instrucţiunile condiţionale, contribuind la efectuarea unor condiţii mai complexe în parantezele instructiunii "if()" si "else if()".

**D. Operatorul condiţional "? :"**

O alta metodă de a executa un cod în funcţie de faptul dacă o expresie este Adevarată sau Falsă e **operatorul "? :"**

Acest operator condiţional, deşi la prima vedere arată diferit de ceilalţi, este o formă prescurtată a instrucţiunii "if( ) else". Sintaxa generală de folosire a lui este:

* **expresie-condiţionala ? dacă -TRUE : dacă -FALSE;**

Operatorul condiţional evaluează expresia condiţională. Dacă expresia are valoarea TRUE, operatorul condiţional returnează valoarea de la "**daca-TRUE**"; în caz contrar, returnează valoarea de la "**daca-FALSE**".

Pe lângă atribuirea unei valori în funcţie de rezultatul unei expresii, acest operator condiţional poate fi utilizat în aceeaşi forma şi la determinarea apelării unei anumite funcţii, după rezultatul unei expresii logice.

* **(expresie-logică) ? dacăTrue( ) : dacăFalse( );**

unde "dacăTrue( )" şi "dacăFalse( )" pot fi considerate doua funcţii.

* 1. **EXEMPLE**

1. ***Sistemul de două ecuaţii de gradul I (varianta C)***

*#include <conio.h>*

*#include <stdio.h>*

*void main(){*

*/\* Rezolvarea sistemului liniar de doua ecuatii cu doua necunoscute*

*ax+by=c*

*dx+ey=f \*/*

*double a,b,c,d,e,f,x,y,det,det1,det2;*

*printf("\n\nIntroduceti coeficientii a,b,c,d,e,f\n");*

*/\* Coeficientii se introduc pe aceeasi linie, separati de blancuri \*/*

*if (scanf("%lf %lf %lf %lf %lf %lf",&a, &b, &c, &d, &e, &f) != 6)*

*printf("Coeficienti eronati\n");*

*else*

*if ((det=a\*e-b\*d) == 0)*

*printf("Sistemul are determinantul nul\n");*

*else*

*{*

*det1=c\*e-b\*f;*

*det2=a\*f-c\*d;*

*x=det1/det;*

*y=det2/det;*

*printf("x=%g\ty=%g\n",x,y);*

*}*

*getch();*

*}*

1. ***Sistemul de două ecuaţii de gradul I (varianta C++)***

*#include <conio.h>*

*#include <fstream.h>*

*void main(){*

*clrscr();*

*float a, b, c, d, e, f, dd, dx, dy, x, y;*

*cout<<"a=";cin>>a;*

*cout<<"b=";cin>>b;*

*cout<<"c=";cin>>c;*

*cout<<"d=";cin>>d;*

*cout<<"e=";cin>>e;*

*cout<<"f=";cin>>f;*

*dd=a\*e - b\*d;*

*if(a\*e - b\*d == 0) cout<<"date improprii";*

*else{*

*dx=c\*e - b\*f;*

*dy=a\*f - c\*d;*

*x=dx/dd;*

*y=dy/dd;*

*cout<<endl;*

*cout<<"x="<<x<<endl;*

*cout<<"y="<<y<<endl;*

*}*

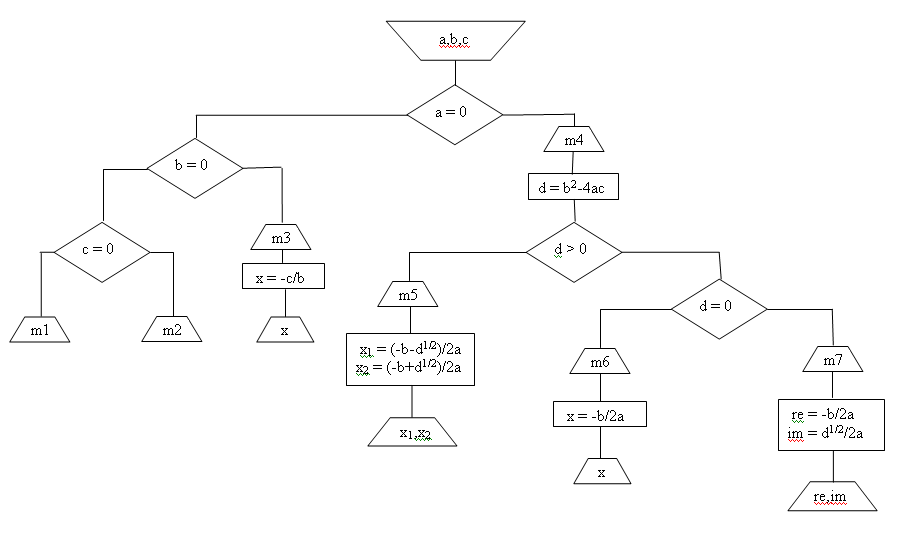
*getch();*

*}*

1. ***Ecuaţia de gradul II***

Să se rezolve ecuaţia de gradul II , coeficienţii *a*, *b* şi *c* fiind citiţi de la tastatură.

Identificând toate situaţiile care pot să apară, obţiem organigrama



În care semnificaţia mesajelor este următoarea:

m1 : Aceasta nici macar nu e o ecuatie!

m2 : Imposibil!

m3 : Este, de fapt, o ecuatie de gradul I

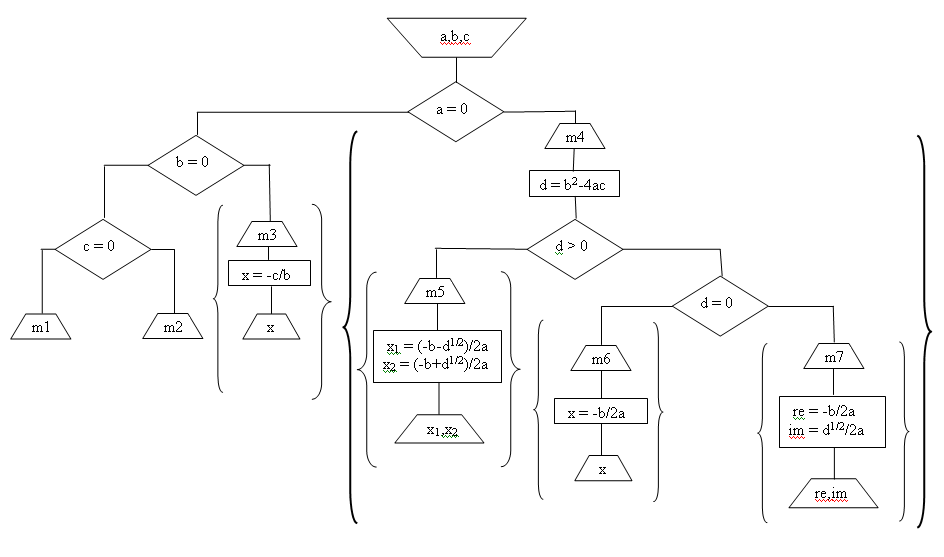
m4 : Asta da! Este o ecuatie de gradul II

m5 : Radacini reale distincte

m6 : Radacini reale confundate

m7 : Radacini complexe

Prin identificarea modului în care instrucţiunile IF-THEN, respectiv IF-THEN-ELSE se include unele pe altele şi a instrucţiunilor simple şi compuse, se obţine organigrama mai explicită:



Codul corespunzător este:

*#include <conio.h>*

*#include <iostream.h>*

*#include <math.h>*

*void main(){*

*float a, b, c, d, x, x1, x2, re, im;*

*clrscr();*

*cout<<"a=";cin>>a;*

*cout<<"b=";cin>>b;*

*cout<<"c=";cin>>c;*

*if(a==0) if(b==0) if(c==0) cout<<"Aceasta nici macar nu e o ecuatie!";*

*else cout<<"Imposibil!";*

*else{*

*cout<<"Este, de fapt, o ecuatie de gradul I"<<endl;*

*x=-c/b;*

*cout<<"Radacina este x="<<x;*

*}*

*else{*

*cout<<"Asta da! Este o ecuatie de gradul II"<<endl;*

*d=b\*b-4\*a\*c;*

*if(d>0) {*

*cout<<"Radacini reale distincte"<<endl;*

*x1=(-b-sqrt(d))/2/a;*

*x2=(-b+sqrt(d))/2/a;*

*cout<<"x1="<<x1<<endl;*

*cout<<"x2="<<x2;*

*}*

*else if(d==0){*

*cout<<"Radacini reale confundate"<<endl;*

*x=-b/2/a;*

*cout<<"x1=x2="<<x;*

*}*

*else{*

*cout<<"Radacini complexe"<<endl;*

*re=-b/2/a;*

*im=-sqrt(-d)/2/a;*

*if(im<0)im=-im;*

*cout<<"x1="<<re<<" - "<<im<<" i"<<endl;*

*cout<<"x2="<<re<<" + "<<im<<" i";*

*}*

*}*

*getch();*

*}*

1. ***Realizarea unui meniu***

Să se realizeze “scheletul” unui meniu pentru lucrul ulterior cu listele înlănţuite; programul va permite selectarea dintr-un meniu a unor acţiuni şi va afişa acţiunea viitoare, pe care programul o va efectua în momentul în care vor fi implementate funcţiile respective.

Meniul va oferi ca principale opţiuni crearea listei, listarea de la stanga la dreapta, listarea de la dreapta la stanga, inserarea unui nod in interiorul listei, inserarea unui nod pe prima poziţie, inserarea unui nod pe ultima poziţie, ştergerea unui nod din interiorul listei, ştergerea primului nod, ştergerea ultimului nod şi salvarea listei.

Codul corespunzător este:

*#include <iostream.h>*

*#include <conio.h>*

*#include <stdlib.h>*

*void main(){*

*int rasp;*

*clrscr();*

*cout<<"OPERATII CU LISTE :"<<endl<<endl;*

*cout<<*

*" 1: creare"<<endl<<*

*" 2: listare de la stanga la dreapta"<<endl<<*

*" 3: listare de la dreapta la stanga"<<endl<<*

*" 4: inserare nod in interiorul listei"<<endl<<*

*" 5: inserare nod pe prima pozitie"<<endl<<*

*" 6: inserare nod pe ultima pozitie"<<endl<<*

*" 7: stergerea unui nod din interiorul listei"<<endl<<*

*" 8: stergerea primului nod"<<endl<<*

*" 9: stergerea ultimului nod"<<endl<<*

*"10: salvez lista"<<endl;*

*cin>>rasp;*

*switch (rasp){*

*case 1: cout<<"voi crea o lista";break;*

*case 2: cout<<"voi lista de la stanga la dreapta";break;*

*case 3: cout<<"voi lista de la dreapta la stanga";break;*

*case 4: cout<<" voi insera un nod in interiorul listei";break;*

*case 5: cout<<" voi insera un nod inaintea primului nod";break;*

*case 6: cout<<" voi insera un nod dupa ultimul nod";break;*

*case 7: cout<<" voi sterge un nod din interiorul listei";break;*

*case 8: cout<<" voi sterge primul nod";break;*

*case 9: cout<<" voi sterge ultimul nod";break;*

*case 10: cout<<" voi salva lista";break;*

*default: exit(1);*

*}*

*getch();*

*}*

* 1. **TEME DE LABORATOR**

1. Se citesc de la tastatură patru numere întregi ***a***, ***b***, ***c*** şi ***d***, toate diferite. Să se afişeze cea mai mare valoare citită.
2. Se citesc de la tastatură patru numere întregi ***a***, ***b***, ***c*** şi ***d***, toate diferite. Să se afişeze cea mai mică valoare citită.
3. Se citesc de la tastatură trei numere întregi ***a***, ***b*** şi ***c***, nu neaparat diferite. Să se afişeze numele varibilei care are cea mai mare valoare. Daca mai multe variabie au această valoare, se vor tipări numele tuturor variabilelor care au această valoare, pe linii diferite ale monitorului.
4. Se citeşte de la tastatură un număr natural ***n***. Să se afişeze mesajul “numărul este par”, sau mesajul “numărul este impar”, în funcţie de paritatea numărului ***n***.
5. Se citesc de la tastatură două numere naturale ***n*** şi ***k***. Să se afişeze mesajul “numărul n este divizibil la numărul k”, sau mesajul “numărul n NU este divizibil la numărul k”, în funcţie de valorile introduse.
6. Se citeşte de la tastatură un număr natural ***n***, mai mare ca 100. Să se afişeze, în ordine, pe aceeaşi linie, cifra sutelor, cifra zecilor şi cifra unităţilor numărului n, separate de câte un spaţiu.
7. Să se rezolve ecuaţia de ordinul I ***ax+b=0***, cu verificarea consistenţei valorilor introduse de la tastatură pentru parametrii ***a*** şi ***b***.
8. Să se rezolve ecuaţia de gradul al II-lea, luând în considerare toate posibilităţile pentru valorile introduse de la tastatură pentru paramtrii ***a***, ***b*** şi ***c***.

**BONUS**

1. Se citeşte de la tastatură un număr natural ***n***, mai mare decât 100 şi mai mic decât 65000. Să se elimine din număr cifra (cifrele) din mijloc şi să se afişeze numărul obţinut. \*

\* Dacă numărul ***n*** are un număr impar de cifre, se elimină o singură cifră, iar dacă numărul ***n*** are un număr par de cifre, se elimină două cifre.