Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Informatica si Ingineria Sistemelor

**RAPORT**

Lucrarea de laborator nr.3

la Programarea Calculatoarelor

A efectuat:

st. gr. C-171 D. Melniciuc

A verificat:

dr., conf.univ. M. Kulev

Chişinău 2017

**Lucrarea de laborator nr.3**

**Tema: Programe ciclice**

**Scopul lucrarii practice:**

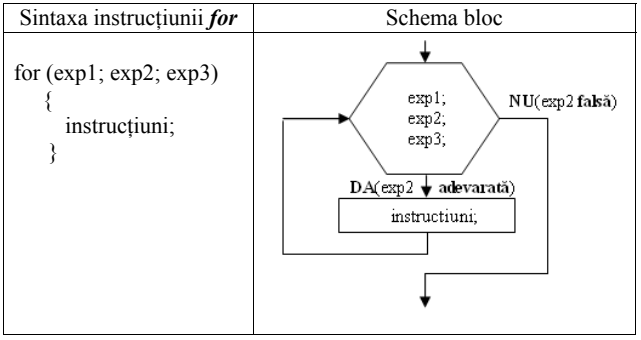
Studierea, utilizarea şi obţinerea deprinderilor practice de

elaborare şi depanare a programelor ramificate (if-else, switch-

break).

**Notiuni de baza:**

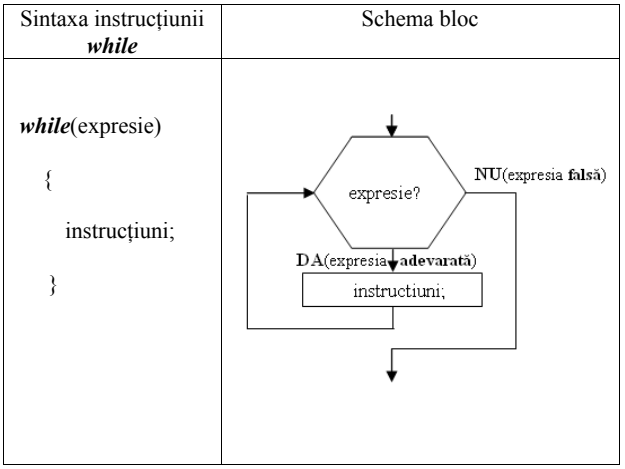
Una dintre cele mai puternice instrucţiuni iterative ale limbajului C/C++ (şi nu numai) este instrucţiunea for, care are urmatoarea sintaxă:



Modul de funcţionare: Procesul de execuţie a ciclului începe cu calcularea expresiei de iniţializare (exp1), după care se trece la expresia de control (exp2) şi se verifică veridicitatea acesteia. În cazul când aceasta e adevărată, se trece la executarea instrucţiunilor din interiorul corpului lui for. După îndeplinirea acestora se trece la expresia de incrimentare/decrementare (exp3) şi din nou se controlează expresia de control (exp2), justeţea căreia duce la îndeplinirea repetată a ciclului. Instrucţiunea for va cicla de atâtea ori, până când expresia de control va fi adevărată. În caz

contrar se va trece la următoarea instrucţiune după corpul lui for.

Ciclul while este utilizat în cazul în care nu se cunoaşte

numărul de repetări ale ciclului şi nu există necesitatea ca ciclul să fie executat cel puţin o singură dată. Sintaxa instrucţiunii while este următoarea****

Modul de funcţionare: Mai întâi se evaluează „expresie”,

dacă aceasta este adevărată, atunci se execută toate instrucţiunile

din corpul ciclului şi controlul trece la începutul buclei while.

Astfel, corpul se execută ori de câte ori „expresie” se evaluează la

"adevăr". Terminarea buclei are loc când „expresie” ia valoarea

fals.

**Condiţia problemei (sarcina de lucru):**

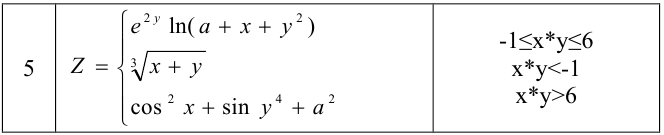
Crearea programului pentru rezolvarea corecta a sistemului de ecuatii prin 3 metode diferite:

-Metoda if

-Metoda if-else

-Metoda if-else-if

**Varianta 5:**



**Mersul lucrarii:**

Pentru a programa la calculator e necesara urmarirea unui algoritm strict. Pasii sunt urmatorii:

-cunoasterea problemei;

Citirea si intelegerii corecte a problemei.

-analiza datelor;

Studierea datelor problemei pentru a destinge datele de tip input, output si date intermediare.

-elaborarea algoritmului prin schema logica;

Elaborarea schemei logice a programului.

-codul (textului) programei;

Scrierea codului propriu-ziz.

-testarea;

Rularea problemei pentru a determina erori logice sau de sintaxa.

-verificarea rezultatelor

Introducerea diferitor date pentru a verifica mai multe posibilitati.

-analiza rezultatelor si formarea concluziilor;

Formare concluziilor adecvate dupa efectuarea pasilor anteriori.

Analiza datelor:

In problema data am urmatoarele date: a, y, x, xmin ,xmax ,dx, Z;

a, y sunt parametrii datelor expresiei

xmin este valoarea minima a lui x

xmax este valoarea maxima a lui x

dx este pasul

Z este valoarea expresiei

**Schema logica**

***Legenda:***

=Nu sunt solutii;

Cond1=-1

Cond2=xy

Cond2=xy>6;

xmin; xmax; dx

Introduceti cazul

Introduceti necunoscutele

Metoda while

Metoda for

1

k

k = 1

2

“Z=..”, Z

Rezultat

Cond1

K=1

1

Cond1

Exp1

Exp1

Exp1

Exp3

Exp2

Cond2

Cond1

Exp3

Stop

**`**

Cond2

2

**Codul (textul) programului in limbajul C:**

#include <stdio.h>

#include <conio.h>

#include <math.h>

#include <stdlib.h>

int main()

{

float a, y, xmin ,xmax ,dx;

float x, Z;

int n; int k;

printf("Introduceti necunoscutele: a, y\n" );

scanf("%f%f", &a, &y);

printf("Introduceti limitele si pasul: x\_min, x\_max, dx\n" );

scanf("%f%f%f", &xmin, &xmax, &dx);

printf("\n1:Metoda while\n");

printf("2:Metoda for\n");

printf("Introduceti cazul (1 - 2)\n\r");

scanf("%d", &k);

switch(k)

{

case 1:

{

x=xmin;

n=0;

while(x<=xmax)

{

if((-1<=x\*y) && (x\*y<=6))

Z=exp(2\*y)\*log(a+x+pow(y,2));

else

{

if(x\*y<-1)

Z=pow(x+y,1/3);

else Z=cos(x)\*cos(x)+pow(sin(y),4)+a\*a;

}

n=n+1;

printf("\n%d. x=%.3f Z=%.3f", n, x, Z);

x=x+dx;

}

break;

}

case 2:

{

n=0;

for(x=xmin; x<=xmax; x=x+dx)

{

if((-1<=x\*y) && (x\*y<=6))

Z=exp(2\*y)\*log(a+x+pow(y,2));

else

{

if(x\*y<-1)

Z=pow(x+y,1/3);

else Z=cos(x)\*cos(x)+pow(sin(y),4)+a\*a;

}

n=n+1;

printf("\n%d. x=%.3f Z=%.3f", n, x, Z);

}

break;

}

default: {printf( "\nNu exista asa caz!\nValoarea lui Z nu a fost calculata\nIntroduceti un caz valid de la 1 la 2" );}

}

return 0;

}

**Metodele folosite in program:**

1: Metoda while

2: Metoda for

**Valorile date parametrilor a, y, xmin, xmax, dx**

a = 1; y = 2;

xmin = 12; xmax = 21;

dx = 3;

**Rezultatele obtinute:**

1. x = 12.000 z = 2.396

2. x = 15.000 z = 2.261

3. x = 18.000 z = 2.120

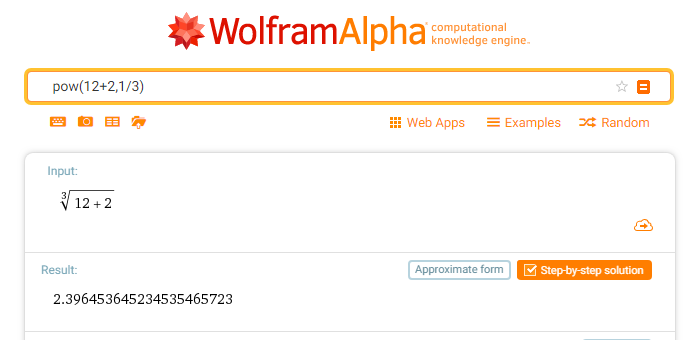
4. x = 21.000 z = 1.984

**Verificarea rezultatelor:**

Pentru verificarea corectitudinii ecuatiilor folosim aplicație WolframAlpha online[2]:

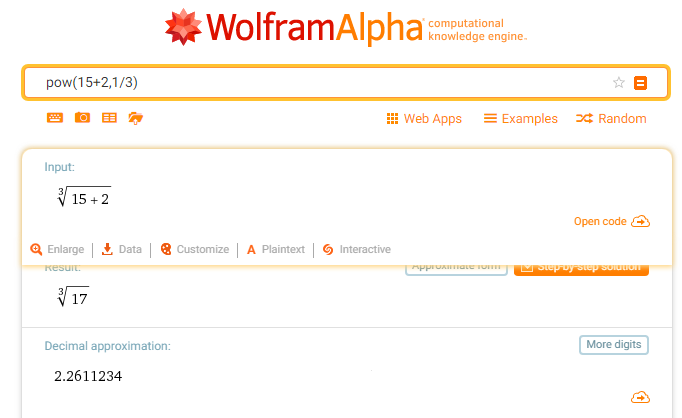


Datele de la compilator



Wolfram:

**X = 12**

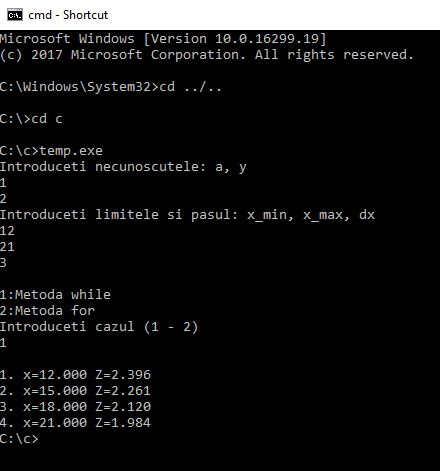


**X = 15**

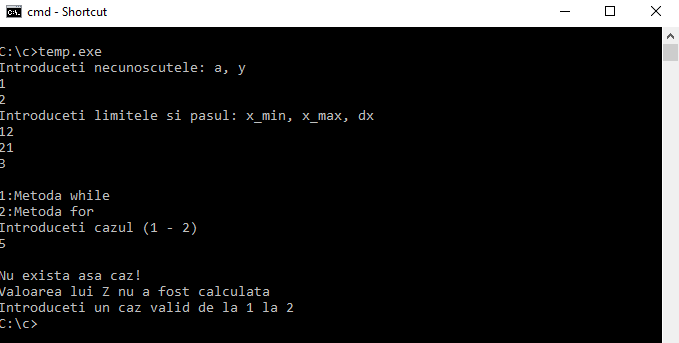
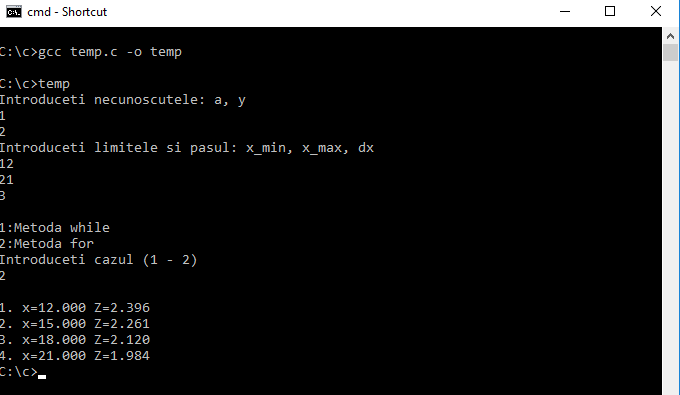


**X = 18**

**Metoda while**



**Metoda for**



**Metoda inexistenta (default)**

**Bibliografie:**

[1]Wolfram-Alfa online

[2]Conspectul si notele de la lectiile teoretice

**Concluzie:**

In urma efectuarii programului si a rularii acestuia au fost obtinute anumite abilitati in domeniu programarii si anume studierea programelor ciclice si lucrarea cu functiile while si for;

**Avantaje**

Algoritmul ramificat izoleaza problema cu mai multe alternative.

Exista comunicare indirecta a progamatorului si a clientului datorita faptului ca programul e interactive.

Putem alege metada preferata de rezolvare a problemei.

Putem da pas lui x pentru a obtine mai multe valori a expresiei matematice.

**Dezavantje**

Nu putem schimba pasul dupa compilarea programului

**Dezvoltarea programului**

Includerea meniului si switch-ului intr-un ciclu.

Verificarea deomeniului de valori a expresiilor matematice intr-un ciclu aparte.