Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova   
  
Universitatea Tehnică a Moldovei   
  
  
  
  
  
  
  
RAPORT

Lucrarea de laborator nr. 2   
la Programarea Calculatoarelor

A efectuat: st. gr. TI-214 Buza Cătălin  
  
A verificat: Prijilevschi Dumitru

UTM, Chișinău 2021

#include <stdio.h>

int main()

{

//Laboratorul nr.2 partea 1 din 3//

printf("Lucrare de laborator nr.2\n");

printf("\nProbleme pentru instructiuni de ramificare:");

printf("\nProblema nr.1:\nScrieti un program C pentru a gasi maxim intre doua numere.");

double a=2.2,b=3.123;

if(a>b) printf("\n%f este maxim",a);

else printf("\n%f este maxim",b);

printf("\nProblema nr.2:\nScrieti un program C pentru a gasi maxim intre trei numere. ");

double x=123,y=123.4,z=123.6,max=0;

if (x>y) printf("\nMaximul primelor 2 numere este:=%f",max=x);

else printf("\nMaximul primelor 2 numere este:=%f",max=y);

if(max>z) printf("\nMaximul maximului primelor 2 numere si al 3 este:=%f",max=max);

else printf("\nMaximul maximului primelor 2 numere si al 3 este:=%f",max=z);

printf("\nProblema nr.3:\nScrieti un program C pentru a verifica daca un numar este negativ, pozitiv sau zero");

int c=-3;

if (c>0) printf("\nNr %d este pozitiv",c);

else if(c<0) printf("\nNr %d este negativ",c);

else printf("\nNr %d este egal cu 0",c);

printf("\nProblema nr.4:\nScrieti un program C pentru a verifica daca un numar este divizibil cu 5 și 11 sau nu. ");

int a1=55;

printf("\nNumarul nostru este:=%d",a1);

if ((a1%5==0)&&(a1%11==0)) printf("\nNumarul nostru e divizibil cu 5 si 11");

else printf("\nNumarul nostru nu e divizibil cu 5 si 11");

printf("\nProblema nr.5:\nScrieti un program C pentru a verifica daca un numar este par sau impar. ");

int b1=12345678;

if(b1%2==0) printf("\nNumarul %d este par",b1);

else printf("\nNumarul %d este impar",b1);

printf("\nProblema nr.6:\nScrieti un program C pentru a verifica daca un an este bisect sau nu. ");

int an=2021;

if ((an%4==0)&&(an%100!=0)||an%400==0) printf("\nAnul %d este bisect!",an);

else printf("\nAnul %d nu e bisect!",an);

printf("\nProblema nr.7:\nScrieti un program C pentru a verifica daca un caracter este sau nu alfabet. ");

char c1='B';

if((c1>='a')&&(c1<='z')) printf("\nCaracterul %c este din alfabet si este o litera mica",c1);

else if((c1>='A')&&(c1<='Z')) printf("\nCaracterul %c este o litera mare din alfabet",c1);

else printf("\nCaracterul %c nu este din alfabet",c1);

printf("\nProblema nr.8:\nScrieti un program C pentru a introduce orice alfabet si verificati dacă este vocala sau consoana. ");

char litera='u';

if((litera=='a')||(litera=='e')||(litera=='i')||(litera=='o')||(litera=='u')) printf("\nLitera ,,%c'' este o vocala ",litera);

else printf("\nLitera ,,%c'' este o consoana",litera);

printf("\nProblema nr.9:\nScrieti un program C pentru a introduce orice caracter si verificati daca este alfabet, cifra sau caracter special. ");

char v1='\*';

if (((v1>='a')&&(v1<='z'))||((v1>='A')&&(v1<='Z'))) printf("\nCaracterul ,,%c'' este alfabet",v1);

else if ((v1>='0')&&(v1<='9')) printf("\nCaracterul ,,%c'' este cifra",v1);

else printf("\nCaracterul ,,%c'' este un caracter special",v1);

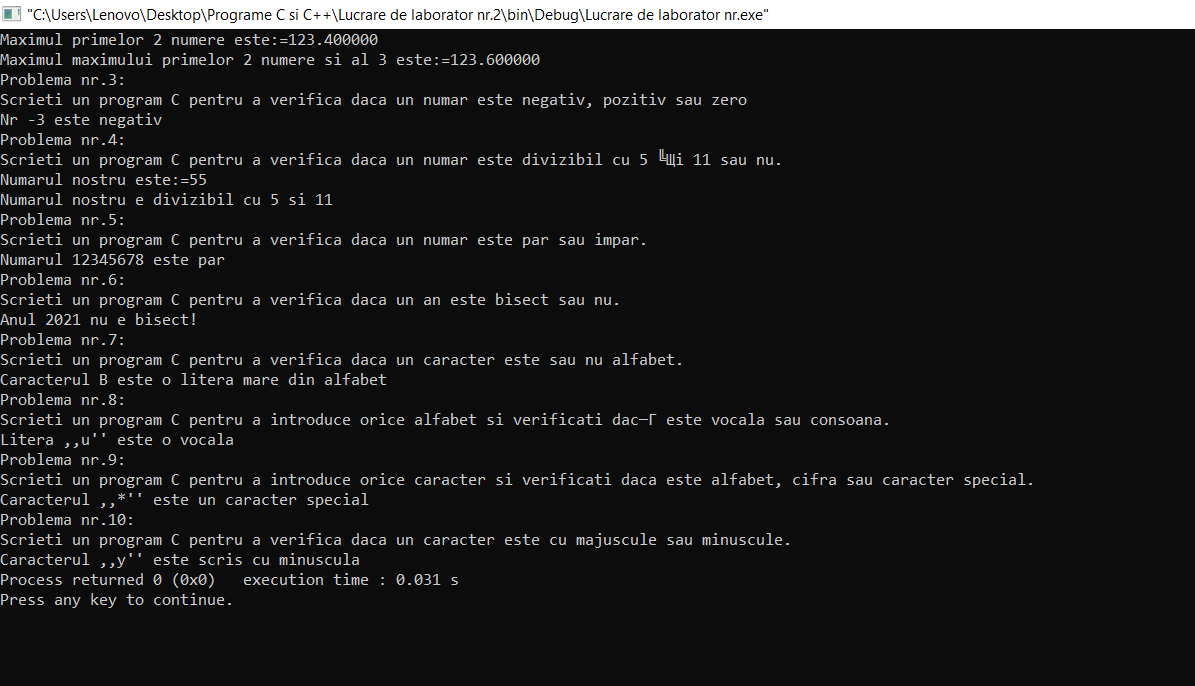
printf("\nProblema nr.10:\nScrieti un program C pentru a verifica daca un caracter este cu majuscule sau minuscule. ");

char m='y';

if ((m>='A') && (m<='Z')) printf("\nCaracterul ,,%c'' este scris cu majuscula",m);

else printf("\nCaracterul ,,%c'' este scris cu minuscula",m);

return 0;

}

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

//Laboratorul nr.2 parte 2 din 3//

printf("Problema nr.11:\nScrieti un program C pentru a introduce numarul saptaminii și a imprima ziua saptaminii. ");

int zi=7;

switch (zi)

{

case 1:printf("\nLuni");break;

case 2:printf("\nMarti");break;

case 3:printf("\nMiercuri");break;

case 4:printf("\nJoi");break;

case 5:printf("\nVineri");break;

case 6:printf("\nSimbata");break;

case 7:printf("\nDuminica");break;

default:

printf("\nNu exista saptamina cu %d zile",zi);

}

printf("\nProblema nr.12:\nScrieți un program C pentru a introduce numărul lunii și a imprima numărul de zile din luna respectivă. ");

int luna=1;

if (luna==1 || luna==3 || luna==5 || luna==7 || luna==8 || luna==10 || luna==12) printf("\nLuna %d are 31 zile",luna);

else if (luna==4 || luna==6 || luna==9 || luna==11) printf("\nLuna %d are 30 de zile",luna);

else if (luna==2) printf("\nLuna %d are 28 sau 29 de zile"); else printf("\nNu exista luna cu %d zile",luna);

printf("\nProblema nr.14\nScrieti un program C pentru a introduce unghiurile unui triunghi și verificati daca triunghiul este valid sau nu. :");

printf("\nIntroducem masura unghiurilor triunghiului nostru:");

double a=55,b=45,c=80;

if (a+b+c==180) printf("\nAcest triunghi este valid");

else printf("\nAcest triunghi nu exista");

printf("\nProblema nr.15:\nScrieti un program C pentru a introduce toate laturile unui triunghi și verificati daca triunghiul este valid sau nu. ");

double d=3,e=4,f=5;

if((d+e>f)&&(d+f>e)&&(e+f>d)) printf("\nTriunghiul cu laturile %lf %lf %lf este valid",d,e,f);

else printf("\nTriunghiul cu laturile %lf %lf %lf nu exista",d,e,f);

printf("\nProblema nr.16:\nScrieti un program C pentru a verifica daca triunghiul este echilateral, isoscel sau triunghi scalen. ");

double x=3,y=4,z=5;

if ((x==y)&&(y==z)) printf("\nTriunghiul este echilateral");

else if ((x==y)&&(y!=z)) printf("\nTriunghiul este isoscel");

else printf("\nTriunghiul este scalen");

printf("\nProblema nr.17:\nScrieti un program C pentru a gasi toate radacinile unei ecuatii patratice.");

printf("\nEcuatia de gradul 2 are forma ax^2+bx+c");

double a1=1,b1=-2,c1=1,delta,x1,x2,s;

delta=pow(b1,2)-4\*a1\*c1;

printf("Ecuatie de gradul 2.\nDelta: b1^2-4\*a1\*c1:= %lf\n",delta);

if(delta>0)

{

x1=(-b1+sqrt(delta))/(2\*a1);

x2=(-b1-sqrt(delta))/(2\*a1);

printf("Delta > 0. Avem 2 solutii reale: x1= %lf si x2= %lf.",x1,x2);}

else if(delta==0)

{

s=(-b1/(2\*a1));

printf("Delta = 0. Avem o singura radacina: s= %lf",s);}

else printf("Delta < 0. Ecuatia are nu are radacini reale");

printf("\nProbleme pentru instructii ciclice");

printf("\nProblema nr.1\nScrieti un program C pentru a imprima toate numerele naturale de la 1 la n. - folosind bucla while.");

unsigned int i=1,n=10;

while(i<=n) {

printf("\n%u ",i);

i++;}

printf("\nProblema nr.2:\nScrieti un program C pentru a imprima toate numerele naturale invers (de la n la 1). - folosind bucla while.");

unsigned int i1=9,n1=1;

while(i1>=n1) {

printf("\n%u",i1);

i1--;}

printf("\nProblema nr.3\nScrieti un program C pentru a imprima toate alfabetele de la a la z. - folosind bucla while");

char m='a',n0='z';

while (m<=n0) {printf("\n%c",m);

m++;}

printf("\nProblema nr.4\nScrieti un program C pentru a imprima toate numerele pare ntre 1 și 100. - utilizind bucla while ");

printf("\nSi problema nr.5\nScrieti un program C pentru a imprima toate numerele impare intre 1 și 100. ");

int i2=1,numar=100;

while (i2<=numar) {if (i2%2==0) printf("\nNumarul %d este par ",i2);

else printf("\nNumarul %d este impar",i2);

i2++;}

printf("\nProblema nr.6\nScrieti un program C pentru a gasi suma tuturor numerelor naturale cuprinse intre 1 și n. ");

int i4=1,sum=0,w=10;

while (i4<=w) {

sum=sum+i4;

i4++;}

printf("\n%d",sum);

printf("\nProblema nr.7:\nScrieti un program C pentru a gasi suma tuturor numerelor pare intre 1 și n.");

printf("\nSi problema nr.8:\nScrieti un program C pentru a gasi suma tuturor numerelor impare cuprinse intre 1 și n.");

int sum1=0,sum2=0,ij,nr=10;

for(ij=1;ij<=nr;ij++)

{

if (ij%2==0) sum1=sum1+ij;

else sum2=sum2+ij;

}

printf("\nSuma numerelor pare de la 1 la %d este:=%d",nr,sum1);

printf("\nSuma numerelor impare de la 1 la %d este:=%d",nr,sum2);

printf("\nProblema nr.9:\nScrieti un program C pentru a imprima tabelul de inmultire a oricarui numar. ");

int k=8,l,prod;

for(l=1;l<=10;l++)

{

prod=k\*l;

printf("\n %dx%d=%d",k,l,prod);

}

printf("\nProblema nr.10:\nScrieti un program C pentru a numara numarul de cifre dintr-un numar. ");

int n2=123456789,n3,n4=n2;

while (n2)

{n2=n2/10;

n3++;

}

printf("\nNr %d are %d cifre",n4,n3);

printf("\nProblema nr.11\nScrieai un program C pentru a gasi prima și ultima cifra a unui numar. ");

int o=4234567,o1,prima\_cifra,ultima\_cifra;

o1=o;

ultima\_cifra=o1%10;

while(o1>10){

o1/=10;

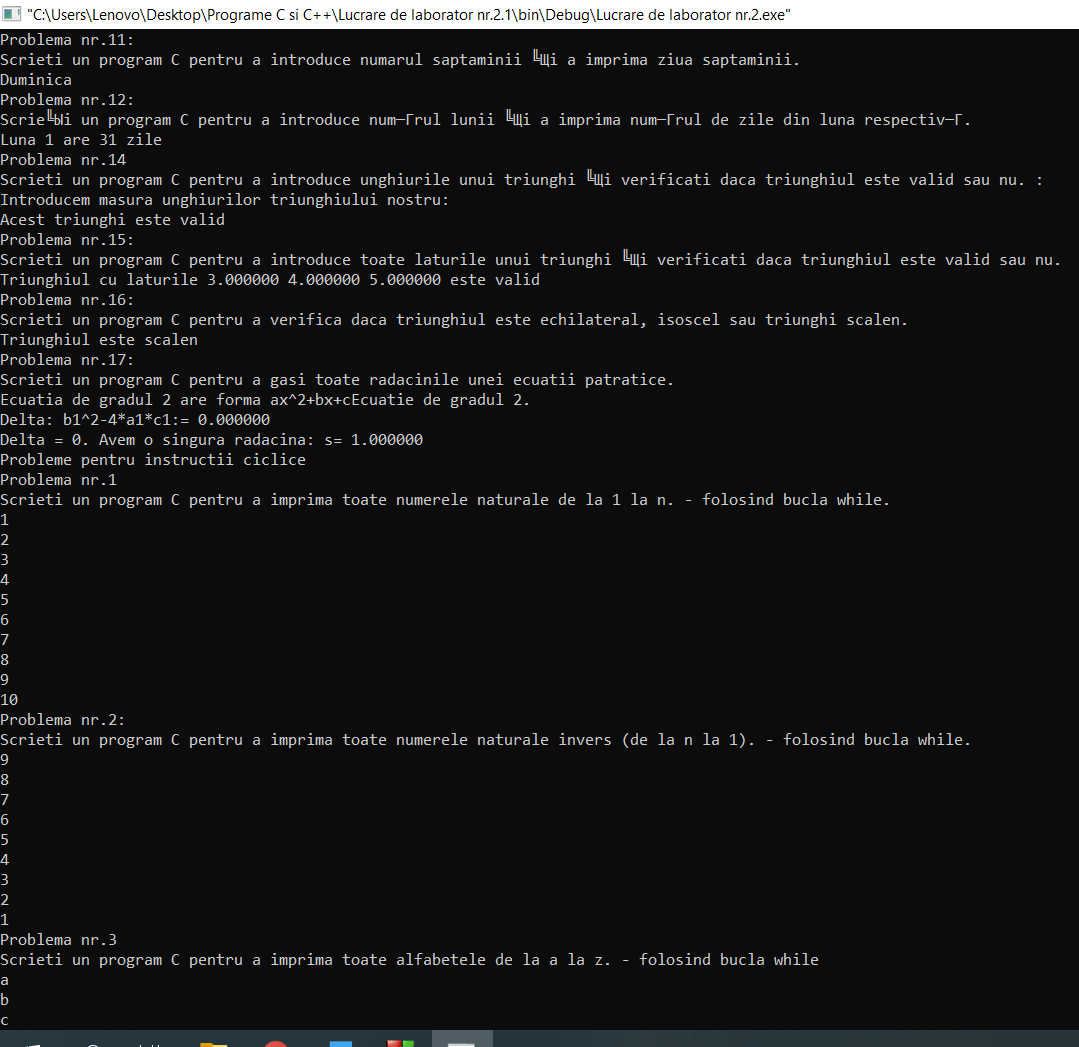
}

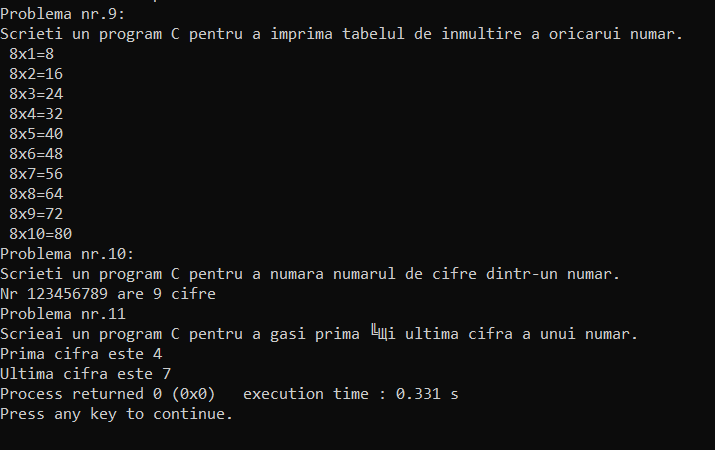
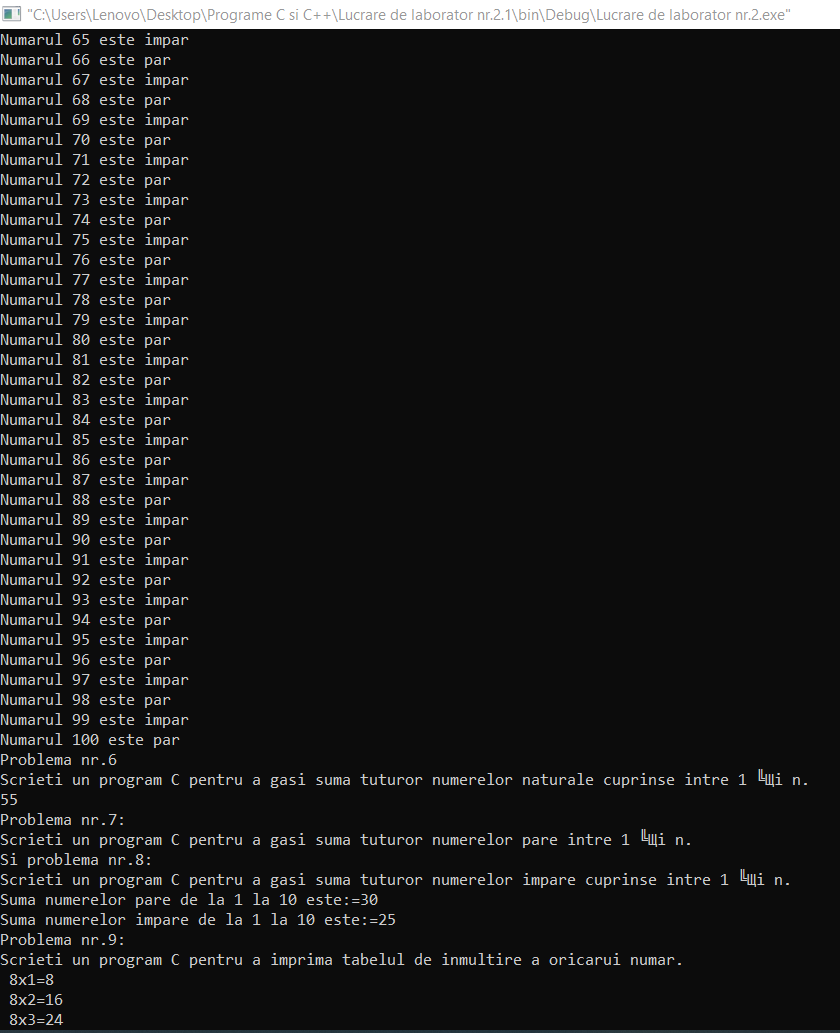
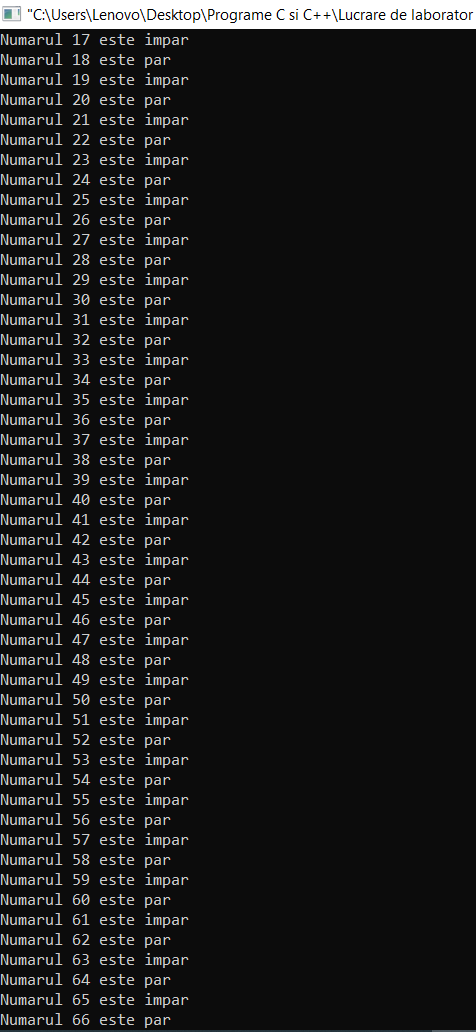
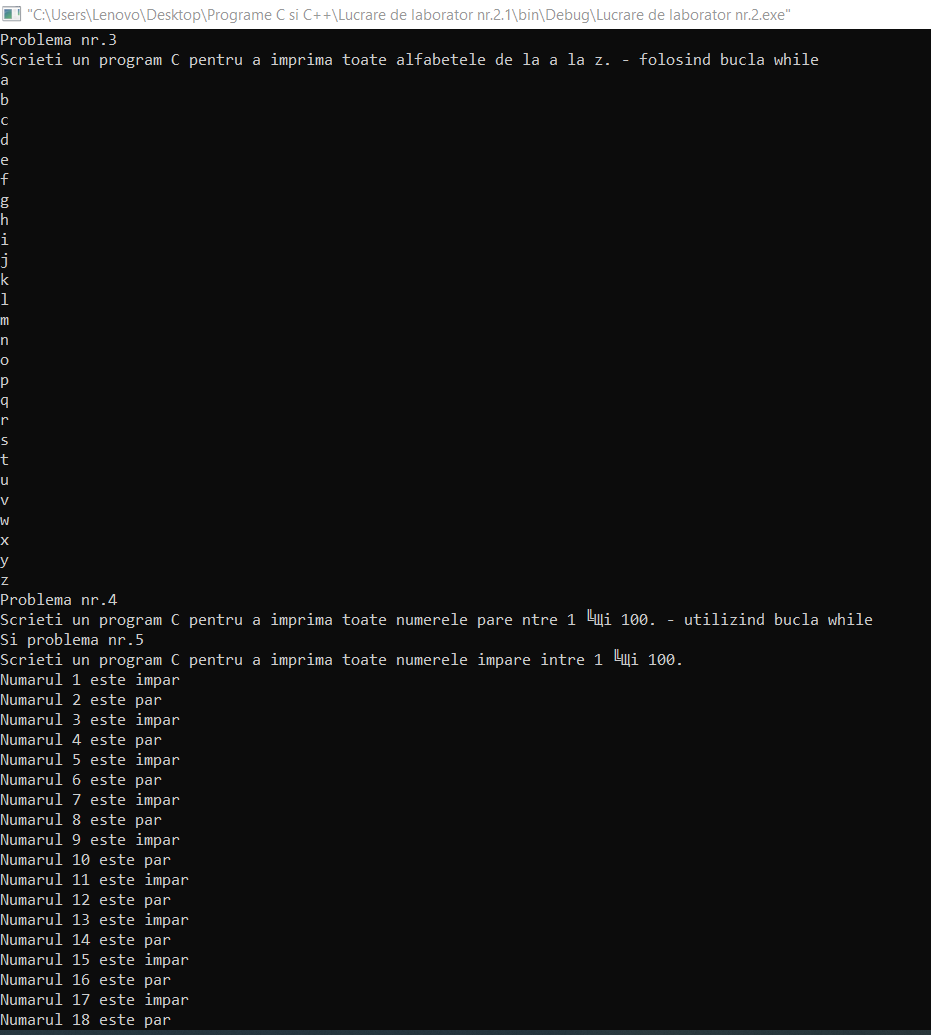
prima\_cifra = o1;

printf("\nPrima cifra este %d",prima\_cifra);

printf("\nUltima cifra este %d",ultima\_cifra);

return 0;

}



#include <stdio.h>

#include <stdlib.h>

int main()

{

//Laboratorul nr.2 partea 3 din 3//

printf("\nProblema nr.12 \nScrieti un program C pentru a gasi suma primei si ultimei cifre a unui numar. ");

int o=4234567,o1,prima\_cifra,ultima\_cifra,sum;

o1=o;

ultima\_cifra=o1%10;

while(o1>10){

o1/=10;

}

prima\_cifra = o1;

sum= prima\_cifra+ultima\_cifra;

printf("\nPrima cifra a nr %d este %d",o,prima\_cifra);

printf("\nUltima cifra a nr %d este %d",o,ultima\_cifra);

printf("\nSuma dintre prima si ultima cifra a nr %d est=%d",o,sum);

printf("\nProblema nr.14\nScrieti un program C pentru a calcula suma cifrelor unui numar. ");

int n=123,s=0,z=n;

while(n!=0)

{

s=s+n%10;

n=n/10;

}

printf("\nSuma cifrelor nr %d este: %d",z,s);

printf("\nProblema nr.15\nScrieti un program C pentru a calcula produsul cifrelor unui numar. ");

int n1=4232,p=1,z1=n1;

while(n1!=0)

{

p=p\*(n1%10);

n1=n1/10;

}

printf("\nProdusul cifrelor nr %d este: %d",z1,p);

printf("\nProblema nr.23\nScrieti un program C pentru a calcula factorialul unui numar. ");

int i,n2=5,facto=1;

for (i=1;i<=n2;i++)

{

facto=facto\*i;

}

printf("\nFactorialul nr %d este %d",n2,facto);

printf("\nProblema nr.26\n Scrieti un program C pentru a verifica daca un numar este sau nu numar prim. ");

int numar=4,i1=2,nr;

for (i1=2;i1<=numar/2;i1++)

{

if (numar%i1==0)

nr++;

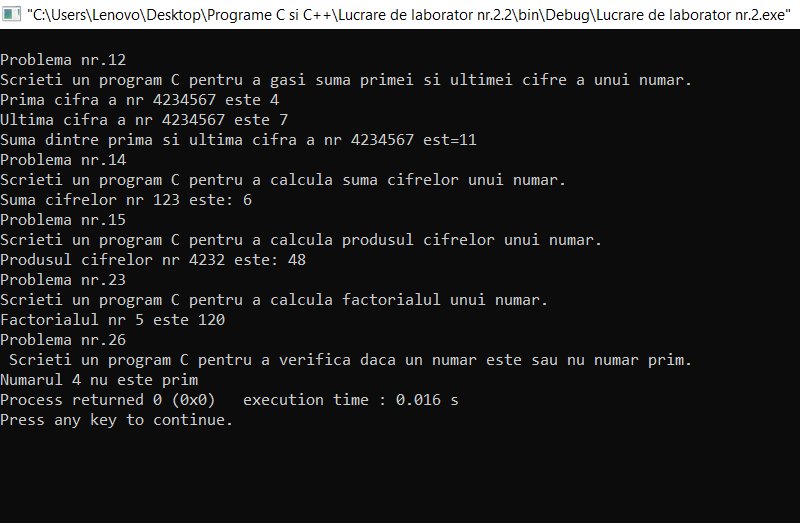
}

if (nr==0) printf("\nNumarul %d este prim",numar);

else printf("\nNumarul %d nu este prim",numar);

return 0;

}



***Concluzii:*** În timpul efectuării lucrării de laborator nr.2 cu tema ,,Instructiuni de ramificare și ciclice ‘’ am atins scopul scopul propus ,cel de a ma familiariza cu instructiunile de ramificare (if,if-else și switch) instrucțiunile ciclice(while,for, do while). O instrucţiune condiţională selectează o singură instrucţiune dintre alternativele sale, pe care apoi o execută. Astfel de instrucţiuni sunt ***if*** ***şi if – else***. Instrucţiunea if e compusă şi sintaxa ei admite unul din următoarele formate:

***if\_(expresie)\_instrucţiune sau***

***if\_(expresie)\_instrucţiune1\_else\_instrucţiune2;***

Instrucţiunea de selectare e destinată pentru selectareaa unei singure variante dintr-o mulţime posibilă de ramificări în program. Orice ramificare poate fi organizată cu ajutorul lanţului de instrucţiuni if\_else\_if\_…\_else\_, însă în cazurile cînd sunt prezente mai mult de 3 condiţii e mai raţională folosirea ***lui switch***. Instrucţiunea de selectare este compusă şi are formatul prezentat mai jos: ***switch (expresie) {***

***case expr\_const1:instrucţiune1;***

***case expr\_const2:instrucţiune2; …***

***case expr\_const\_n:instrucţiune\_n; default:instrucţiune; }***

Instructiunea ***break*** are loc de a revoca terminarea executiei ***switch***.

Ciclul ***FOR*** posedă următoarele caracteristici: numărul de repetări ale ciclului este cunoscut de la începutul executării lui; conducerea ciclului este efectuată cu ajutorul unei variabile de tip int, numită parametrul ciclului, care, în acest proces ciclic primeşte valori consecutive de la valoarea iniţială dată până la valoarea finală dată. Sintaxa instrucţiunii este următoarea: ***for(expresie1;expresie2;expresie3)*** instrucţiune; unde ***expresie1***– expresie de iniţializare a parametrului ciclului***, expresie2***- expresie de control, ***expresie3***- expresie de incrementare/decrementare(corecţie) a parametrului ciclului.

Ciclul while este folosit în cazul, cînd nu este cunoscut numărul de repetări ale ciclului şi nu există necesitatea ca ciclul să fie executat măcar o singură dată. Instrucţiunea de ciclare while are următoarul format: while (expresie) instrucţiune; Instrucţiunea de ciclu while lucrează în felul următor: dacă expresia este adevărată (sau diferită de zero, ce din punct de vedere al limbajului Turbo C este una şi aceiaşi), atunci instrucţiunea se îndeplineşte o dată şi apoi expresia din nou se testează. Această succesiune de acţiuni, ce constă în testarea expresiei şi îndeplinirea instrucţiunei, se repetă periodic pănă ce expresia nu devine falsă(din punct de vedere al limbajului Turbo C devine egală cu zero). Instrucţiunea se numeşte corpul ciclului şi în majoritatea cazurilor reprezintă un bloc, în componenţa căruia intră cîteva instrcţiuni.

Instrucţiunea ciclică DO\_WHILE se foloseşte în cazul cînd numărul de repetări ale ciclului nu-i cunoscut, dar în acelaşi timp e necesar ca ciclul să fie îndeplinit măcar o singură dată. Instrucţiunea de ciclare do\_while are următoarea formă: ***do instrucţiune while(expresie);*** Instrucţiunea do\_while lucrează în felul următor: la început se îndeplineşte instrucţiunea, apoi se calculează valoarea expresiei. Dacă valoarea expresiei este adevărată, atunci instrucţiunea se îndeplineşte din nou, dacă expresia este falsţă, atunci îndeplinirea ciclului se termină.

.