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-211 Popa Cătălin   
  
A verificat: Stratulat Ștefan

UTM, Chișinău 2021

**Probleme pentru instrucțiuni de ramificare:**

1. **Scrieți un program C pentru a găsi maxim între două numere.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

float a,b;

int max;

printf("Dati a,b\n");

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

if (a>b)

{

printf("Max=a\n");

}

else

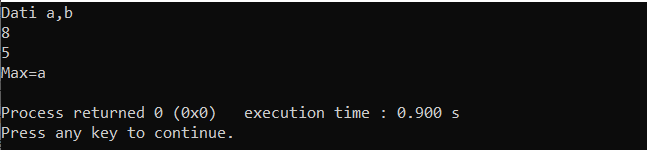
{

printf("Max=b\n");

}

return 0;

}



1. **Scrieți un program C pentru a găsi maxim între trei numere.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

float a,b,c;

int max;

printf("Dati a,b,c\n");

scanf("%f%f%f",&a,&b,&c);

if (b<a&&a>c)

{

printf("Max=a\n");

}

else if (a<b&&b>c)

{

printf("Max=b\n");

}

else

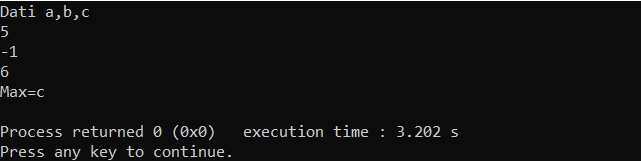
{

printf("Max=c\n");

}

return 0;

}



1. **Scrieți un program C pentru a verifica dacă un număr este negativ, pozitiv sau zero.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

float n;

printf("Dati un numar\n");

scanf("%f",&n);

if (n<0)

{

printf("Numarul este negativ\n");

}

else if (n==0)

{

printf("Numarul este egal cu 0\n");

}

else

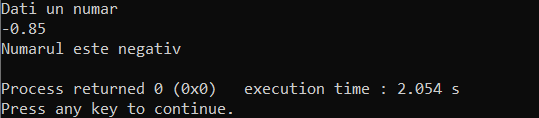
{

printf("Numarul este pozitiv\n");

}

return 0;

}



1. **Scrieți un program C pentru a verifica dacă un număr este divizibil cu 5 și 11 sau nu.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n;

printf("Dati un numar\n");

scanf("%d",&n);

if (n%5==0&&n%11==0)

{

printf("Numarul este divizibil cu 5 si 11\n");

}

else

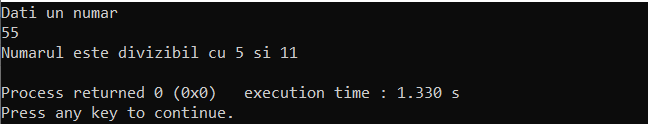
{

printf("Numarul nu este divizibil cu 5 si 11\n");

}

return 0;

}



1. **Scrieți un program C pentru a verifica dacă un număr este par sau impar.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n;

printf("Dati un numar\n");

scanf("%d",&n);

if (n%2==0)

{

printf("Numarul este par\n");

}

else

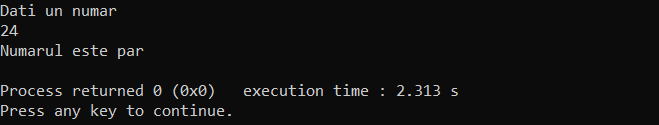
{

printf("Numarul este impar\n");

}

return 0;

}



1. **Scrieți un program C pentru a verifica dacă un an este bisect sau nu.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a;

printf("Dati un an\n");

scanf("%d",&a);

if ((a%400==0)||(a%4==0 && a%100!=0))

{

printf("Este an bisect\n");

}

else

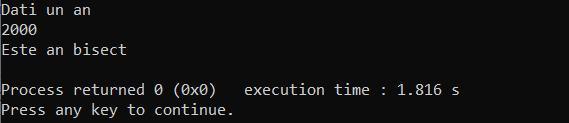
{

printf("Nu este an bisect\n");

}

return 0;

}



1. **Scrieți un program C pentru a verifica dacă un caracter este sau nu alfabet.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

char f;

printf("Introduceti un caracter\n");

scanf("%c",&f);

if (isalpha(f))

{

printf("Caracterul este alfabet\n");

}

else

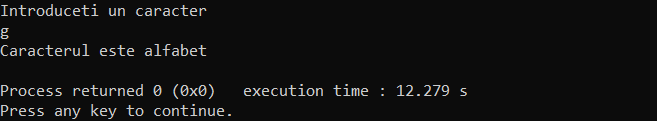
{

printf("Caracterul nu este alfabet\n");

}

return 0;

}



1. **Scrieți un program C pentru a introduce orice alfabet și verificați dacă este vocală sau consoană.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

char f;

printf("Introduceti un caracter din alfabet\n");

scanf("%c",&f);

if (f=='a'||f=='e'||f=='i'||f=='o'||f=='u'||f=='A'||f=='E'||f=='I'||f=='O'||f=='U')

{

printf("Este vocala\n");

}

else

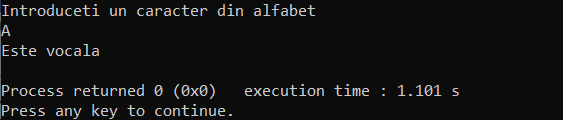
{

printf("Este consoana\n");

}

return 0;

}



1. **Scrieți un program C pentru a introduce orice caracter și verificați dacă este alfabet, cifră sau caracter special.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

char c;

printf("Introduceti un caracter\n");

scanf("%c",&c);

if(c>='0' && c<='9')

{

printf("Caracterul este o cifra\n");

}

else if (isalpha(c))

{

printf("Caracterul face parte din alfabet\n");

}

else

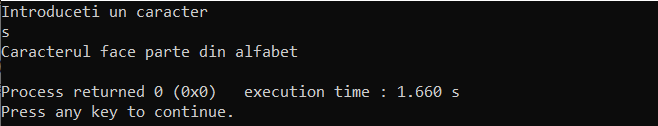
{

printf("Este un caracter special\n");

}

return 0;

}



1. **Scrieți un program C pentru a verifica dacă un caracter este cu majuscule sau minuscule.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

char c;

printf("Introduceti o litera\n");

scanf("%c",&c);

if(c>='A' && c<='Z')

{

printf("Litera este cu majuscula\n");

}

else

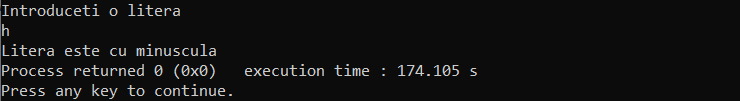
{

printf("Litera este cu minuscula");

}

return 0;

}



1. **Scrieți un program C pentru a introduce numărul săptămânii și a imprima ziua săptămânii.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n;

printf("Introduceti numarul saptamanii \n");

scanf("%d",&n);

if(n==1)

{

printf("Luni\n");

}

else if(n==2)

{

printf("Marti\n");

}

else if(n==3)

{

printf("Miercuri\n");

}

else if(n==4)

{

printf("Joi\n");

}

else if(n==5)

{

printf("Vineri\n");

}

else if(n==6)

{

printf("Sambata\n");

}

else if(n==7)

{

printf("Duminica\n");

}

else

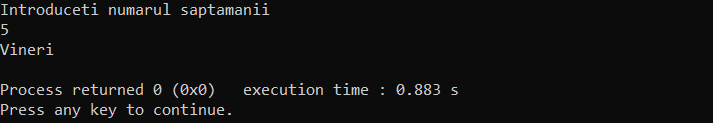
{

printf("Nu exista vre-o saptamana sub numarul respectiv");

}

return 0;

}



1. **Scrieți un program C pentru a introduce numărul lunii și a imprima numărul de zile din luna respectivă.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n;

printf("Introduceti numarul lunii \n");

scanf("%d",&n);

if(n==1)

printf("31 zile \n");

else if(n==2)

printf("28 sau 29 zile \n");

else if(n==3)

printf("31 zile \n");

else if(n==4)

printf("30 zile \n");

else if(n==5)

printf("31 zile \n");

else if(n==6)

printf("30 zile \n");

else if(n==7)

printf("31 zile \n");

else if(n==8)

printf("31 zile \n");

else if(n==9)

printf("30 zile \n");

else if(n==10)

printf("31 zile \n");

else if(n==11)

printf("30 zile \n");

else if(n==12)

printf("31 zile \n");

else

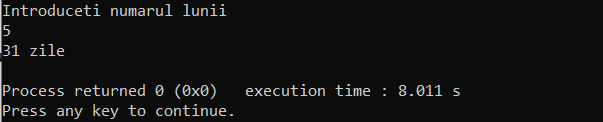
{

printf("Nu exista vre-o luna sub numarul respectiv");

}

return 0;

}



1. **Scrieți un program C pentru a număra numărul total de cifre într-un număr.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

long int n,a=0;

printf("Introduceti numarul \n");

scanf("%ld",&n);

if(n==0)

{

a++;

}

else

{

while(n)

{

a++;

n=n/10;

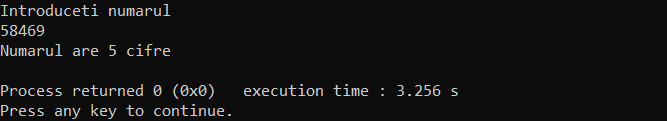
}

}

printf("Numarul are %ld cifre\n",a);

return 0;

}



1. **Scrieți un program C pentru a introduce unghiurile unui triunghi și verificați dacă triunghiul este valid sau nu.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a,b,c;

printf("Introduceti unghiurile triunghiului \n");

scanf("%d%d%d",&a,&b,&c);

if (a+b+c==180)

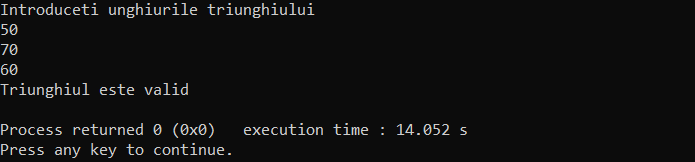
printf("Triunghiul este valid \n");

else

printf("Triunghiul nu este valid \n");

return 0;

}



1. **Scrieți un program C pentru a introduce toate laturile unui triunghi și verificați dacă triunghiul este valid sau nu.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

float a,b,c;

printf("Introduceti laturile triunghiului \n");

scanf("%f%f%f",&a,&b,&c);

if (a+b>=c && a+c>=b && b+c>=a)

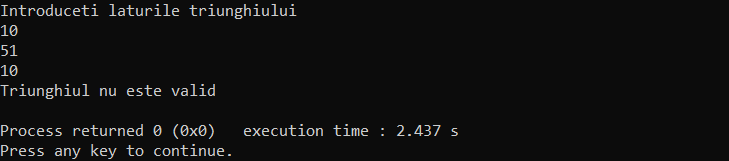
printf("Triunghiul este valid \n");

else

printf("Triunghiul nu este valid \n");

return 0;

}



1. **Scrieți un program C pentru a verifica dacă triunghiul este echilateral, isoscel sau triunghi scalen.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

float a,b,c;

printf("Introduceti lautile triunghiului\n");

scanf("%f%f%f",&a,&b,&c);

if (a==b && b==c && a==c)

printf("Triunghiul este echilateral\n");

else if (a==b || a==c || b==c)

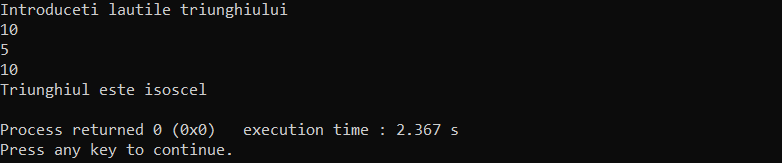
printf("Triunghiul este isoscel\n");

else

printf("triunghiul este scalen\n");

return 0;

}



1. **Scrieți un program C pentru a găsi toate rădăcinile unei ecuații pătratice.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

double a,b,c,d,x1,x2,r,i;

printf("Introduceti termenii ecuatiei\n");

scanf("%lf%lf%lf",&a,&b,&c);

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

if(d>0)

{

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

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

printf("x1= %.3lf, x2= %.3lf",x1,x2);

}

else if(d==0)

{

x1=-b/(2\*r);

printf("x1=x2= %.3lf",x1);

}

else

{

r=-b/(2\*a);

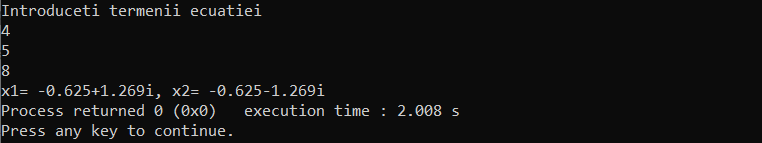
i=sqrt(-d)/(2\*a);

printf("x1= %.3lf+%.3lfi, x2= %.3lf-%.3lfi",r,i,r,i);

}

return 0;

}



**Probleme pentru instrucțiuni ciclice:**

1. **Scrieți un program C pentru a imprima toate numerele naturale de la 1 la n. - folosind bucla while.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i=0;

printf("Introdu un numar natural\n");

scanf("%d",&n);

if(n>0)

{

while(i<=n)

{

printf("%d",i++);

}

}

else

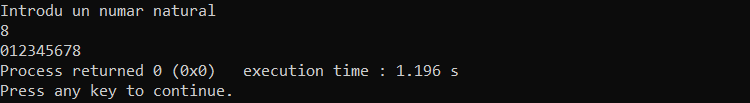
{

printf("Numarul nu corespunde conditiei\n");

}

return 0;

}



1. **Scrieți un program C pentru a imprima toate numerele naturale invers (de la n la 1). - folosind bucla while.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i=1;

printf("Introdu un numar natural\n");

scanf("%d",&n);

if(n>1)

{

while(n>=i)

{

printf("%d",--n);

}

}

else

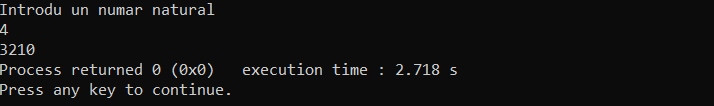
{

printf("Numarul nu corespunde conditiei\n");

}

return 0;

}



**3. Scrieți un program C pentru a imprima toate alfabetele de la a la z. - folosind bucla while.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

char a='a';

while(a<='z')

{

printf("%c",a);

a++;

}

return 0;

}



**4. Scrieți un program C pentru a imprima toate numerele pare între 1 și 100. - utilizând bucla while.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n;

n=2;

while(n<=100)

{

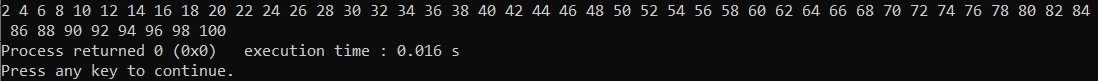
printf("%d ",n);

n=n+2;

}

return 0;

}



**5. Scrieți un program C pentru a imprima toate numerele impare între 1 și 100.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n;

n=1;

while(n<=100)

{

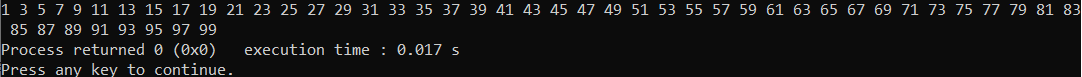
printf("%d ",n);

n=n+2;

}

return 0;

}



**6. Scrieți un program C pentru a găsi suma tuturor numerelor naturale cuprinse între 1 și n.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

unsigned int n,s=0;

printf("Dati un numar\n");

scanf("%d",&n);

for(int i=1;i<=n;i++)

{

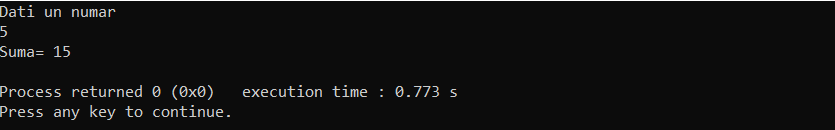
s+=i;

}

printf("Suma= %d\n",s);

return 0;

}



**7. Scrieți un program C pentru a găsi suma tuturor numerelor pare între 1 și n.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

unsigned int n,s=0;

printf("Dati un numar\n");

scanf("%d",&n);

for(int i=2;i<=n;i+=2)

{

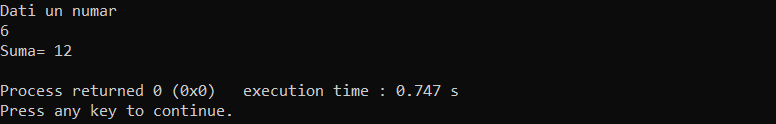
s+=i;

}

printf("Suma= %d\n",s);

return 0;

}



**8. Scrieți un program C pentru a găsi suma tuturor numerelor impare cuprinse între 1 și n.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

unsigned int n,s=0;

printf("Dati un numar\n");

scanf("%d",&n);

for(int i=1;i<=n;i+=2)

{

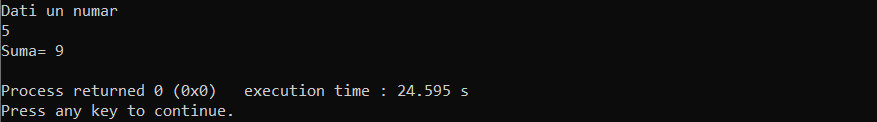
s+=i;

}

printf("Suma= %d\n",s);

return 0;

}



**9. Scrieți un program C pentru a imprima tabelul de înmulțire a oricărui număr.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a;

printf("Dati un numar\n");

scanf("%d",&n);

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

{

printf("%d \* %d= %d\n",n,a,n\*a);

}

return 0;

}



**10. Scrieți un program C pentru a număra numărul de cifre dintr-un număr.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a=0;

printf("Dati un numar\n");

scanf("%d",&n);

while(n>0)

{

n=n/10;

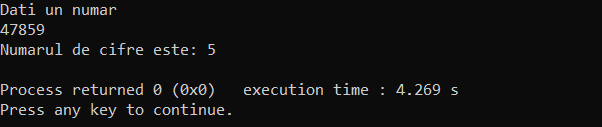
a++;

}

printf("Numarul de cifre este: %d\n",a);

return 0;

}



12.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,a=0,b;

printf("Dati un numar: ");

scanf("%d",&n);

b=n%10;

for(int i=0;i<=n;i++)

{

if(n<10){

break;

}

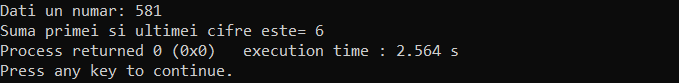
n/=10;

}

printf("Suma primei si ultimei cifre este= %d",b+n);

return 0;

}



**14. Scrieți un program C pentru a calcula suma cifrelor unui număr.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a=0,b ;

printf("Dati un numar:");

scanf("%d",&n);

while(n>0)

{

b=n%10;

n=n/10;

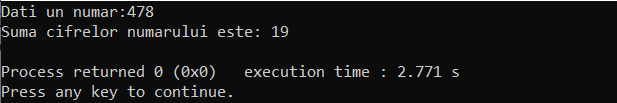
a=a+b;

}

printf("Suma cifrelor numarului este: %d\n",a);

return 0;

}



**15. Scrieți un program C pentru a calcula produsul cifrelor unui număr.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a=1,b ;

printf("Dati un numar:");

scanf("%d",&n);

while(n>0)

{

b=n%10;

n=n/10;

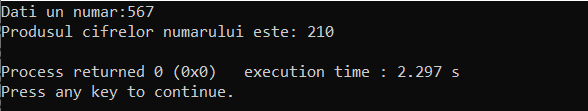
a=a\*b;

}

printf("Produsul cifrelor numarului este: %d\n",a);

return 0;

}



**16. Scrieți un program C pentru a introduce un număr și imprimați invers**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a=0,b;

printf("Dati un numar: ");

scanf("%d", &n);

while (n!= 0)

{

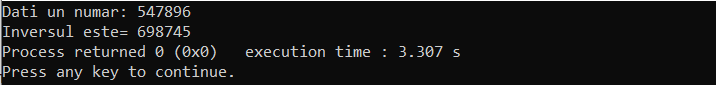
b=n%10;

a=a\*10+b;

n/= 10;}

printf("Inversul este= %d",a);

return 0;}



**17.Scrieți un program C pentru a verifica dacă un număr este palindrom sau nu.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a=0,b,x;

printf("Enter an integer: ");

scanf("%d",&n);

x=n;

while (n!=0) {

b=n%10;

a=a\*10+b;

n/=10;

}

if (x==a)

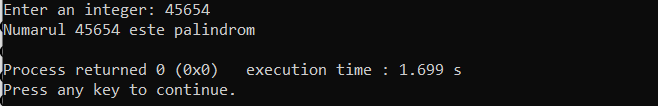
printf("Numarul %d este palindrom \n",x);

else

printf("numarul %d nu este palindrom \n",x);

return 0;

}



**19. Scrieți un program C pentru a introduce un număr și imprimați-l în cuvinte.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a=0;

printf("Dati un numar: ");

scanf("%d",&n);

while(n!= 0)

{

a=(a\*10)+(n%10);

n/=10;

}

while(a!= 0)

{

switch(a%10)

{

case 0:

printf("Zero,");

break;

case 1:

printf("Unu,");

break;

case 2:

printf("Doi,");

break;

case 3:

printf("Trei,");

break;

case 4:

printf("Patru,");

break;

case 5:

printf("Cinci,");

break;

case 6:

printf("Sase,");

break;

case 7:

printf("Sapte,");

break;

case 8:

printf("Opt,");

break;

case 9:

printf("Noua,");

break;

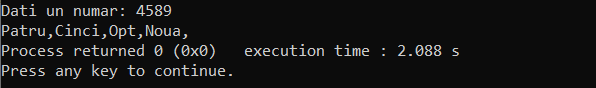
}

a=a/10;

}

return 0;

}



**20. Scrieți un program C pentru a imprima toate caracterele ASCII cu valorile lor.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n;

for(n=0;n<=255;n++)

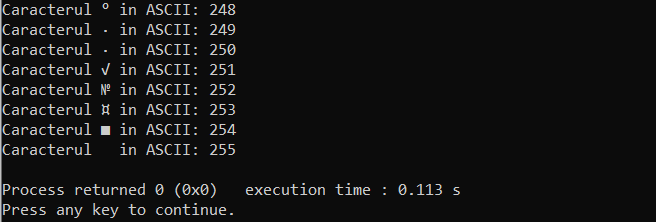
{

printf("Caracterul %c in ASCII: %d \n",n,n);

}

return 0;

}



**32. Scrieți un program C pentru a verifica dacă un număr este sau nu un număr perfect.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int i,n,s=0;

printf("Introduceti un numar: ");

scanf("%d",&n);

for(i=1;i<=n/2;i++)

{

if(n%i==0)

{

s+=i;

}

}

if(s==n && n>0)

{

printf("%d este numar perfect",n);

}

else

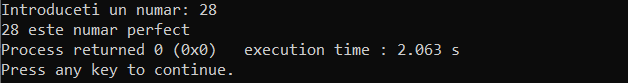
{

printf("%d nu este numar perfect",n);

}

return 0;

}



**36. Scrieți un program C pentru a imprima seria Fibonacci până la n termeni.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,a,b,g;

printf("Dati un numar\n");

scanf("%d",&n);

a=b=1;

for(int i=0;i<=n;i++)

{

g=b+a;

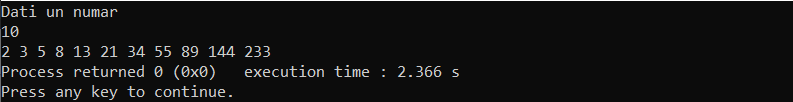
printf("%d ",g);

a=b;

b=g;

}

return 0;}



**45. Scrieți un program C pentru a converti sistemul numeric Zecimal în Binar.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,i,a;

printf("Introdu n: \n");

scanf("%d", &n);

printf("%d in binar:\n", n);

for (i=3; i>=0; i--){

a=n>>i;

if(a&1)

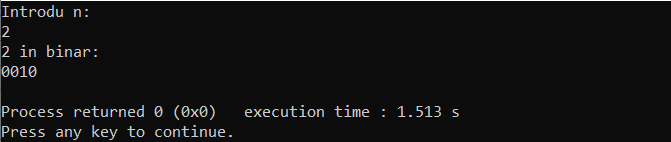
printf("1");

else

printf("0"); }

printf("\n");

return 0;}



**Intrebari**:

* ce instrucțiuni de ramificare sunt în limbajul C?

1. If
2. If-else

* ce instrucțiuni ciclice sunt în limbajul C?

1. For-cu parametru
2. While-procedata de conditie
3. Do-while-cu postconditie

* care este diferența între while și do while?

Ciclul while mai intai verifica valoarea de adevar a conditie, si dupa o executa, iar ciclul do-while mai intai citeste o instructiune , si dupa verifica valoarea de adevar.

* care este diferența între for și while?

In ciclul for este necesar sa fie mentionat intervalul de repetari , iar in while nu este necesar de a cunoaste numarul de repetari.

* care este diferența între if și switch?

Ambele instructiuni sunt folosite pentru a verifica o conditie,dar cand avem o multime de cazuri este mai usor sa folosim switch.

* ce înseamnă break, continue ?

1. Break-intrerupe executarea buclei.
2. Continue-intrerupe doar executarea repetarii curente.

* ce înseamnă contor, condiție de oprire și pasul contorului?

1. Contor-variabila care numara pasii.

Conditie de oprire- limita intervalului la care contorul se opreste.