****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

Achirus Valeria

Gr. IA-203

**Lucrare independenta**

***la cursul de “Programarea calculatoarelor”***

Verificat:

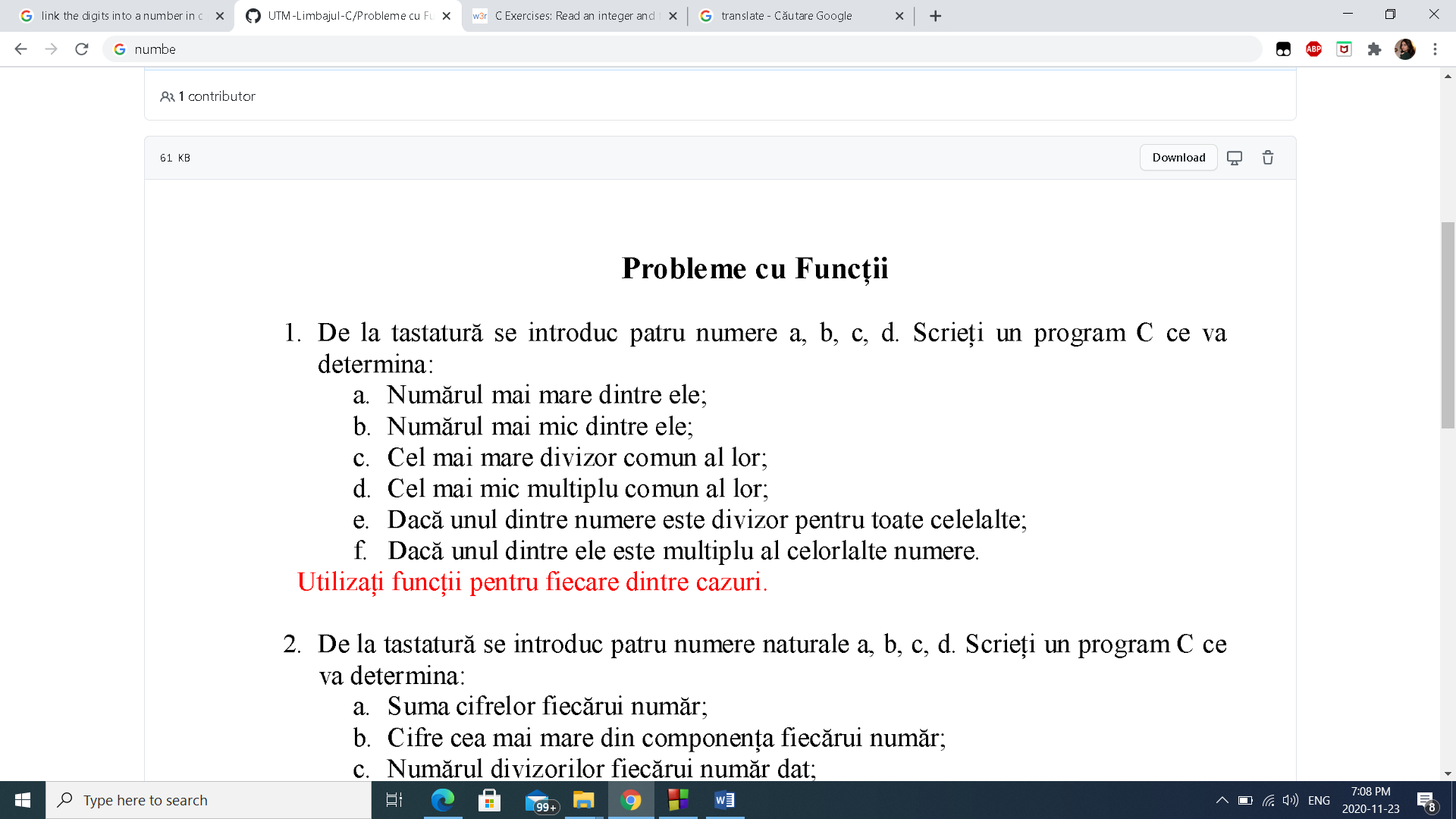
**Guțu Maria,** *doctor, conf. univ.*

Departamentul Informatică şi IS,

Facultatea FCIM, UTM

Chisinau 2020

EX.1



#include <stdio.h>

#include <stdlib.h>

int Max(int x,int y,int z,int w,int max)

{

max=x;

if(y>max){

max=y;

}

if(z>max){

max=z;

}

if(w>max){

max=w;

}

return max;}

int Min(int x,int y,int z,int w,int min)

{

min=x;

if(y<min){

min=y;

}

if(z<min){

min=z;

}

if(w<min){

min=w;

}

return min;}

int Divizor(int x,int y,int z,int w)

{

while (x!=y) {

if (x>y) x=x-y;

else y=y-x;

}

while (y!=z) {

if (y>z) y=y-z;

else z=z-y;

}

while (z!=w) {

if (z>w) z=z-w;

else w=w-z;

}

return w ;}

int Multiplu(int x,int y,int z,int w)

{

int x1=x,y1=y,z1=z,w1=w,multiplu;

while (x1!=y1){

if(x1>y1) x1=x1-y1;

else y1=y1-x1;

}

while (y1!=z1){

if(y1>z1) y1=y1-z1;

else z1=z1-y1;

}

while (z1!=w1){

if(y1>w1) z1=z1-w1;

else w1=w1-z1;

}

multiplu=(x\*y\*z\*w)/w1;

return multiplu;}

int verificare\_Divizor(int x,int y,int z,int w,int min)

{

int m=Min(x,y,z,w,min);

return x%m+y%m+z%m+w%m;

}

int verificare\_multiplu(int x,int y,int z,int w)

{

int max,m=Max(x,y,z,w,max);

return m%x+m%y+m%z+m%w;}

int main()

{

int a,b,c,d;

printf("Dati valoarea a:\n",a);

scanf("%d",&a);

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

scanf("%d",&b);

printf("Dati valoarea a:\n",c);

scanf("%d",&c);

printf("Dati valoarea a:\n",d);

scanf("%d",&d);

int result = Max(a,b,c,d,result);

printf ("Numarul maxim este:%d\n",result);

int result1 = Min (a,b,c,d,result1);

printf ("Numarul minim este:%d\n",result1);

int result2 = Divizor (a,b,c,d);

printf ("Cel mai mare divizor este:%d\n",result2);

int result3 = Multiplu (a,b,c,d);

printf ("Cel mai mic multiplu este:%d\n",result3);

int result4 = verificare\_Divizor(a,b,c,d,result4);

if (result4==0){

printf("Sunt divizori\n");}

else{printf("nu sunt divizori\n");}

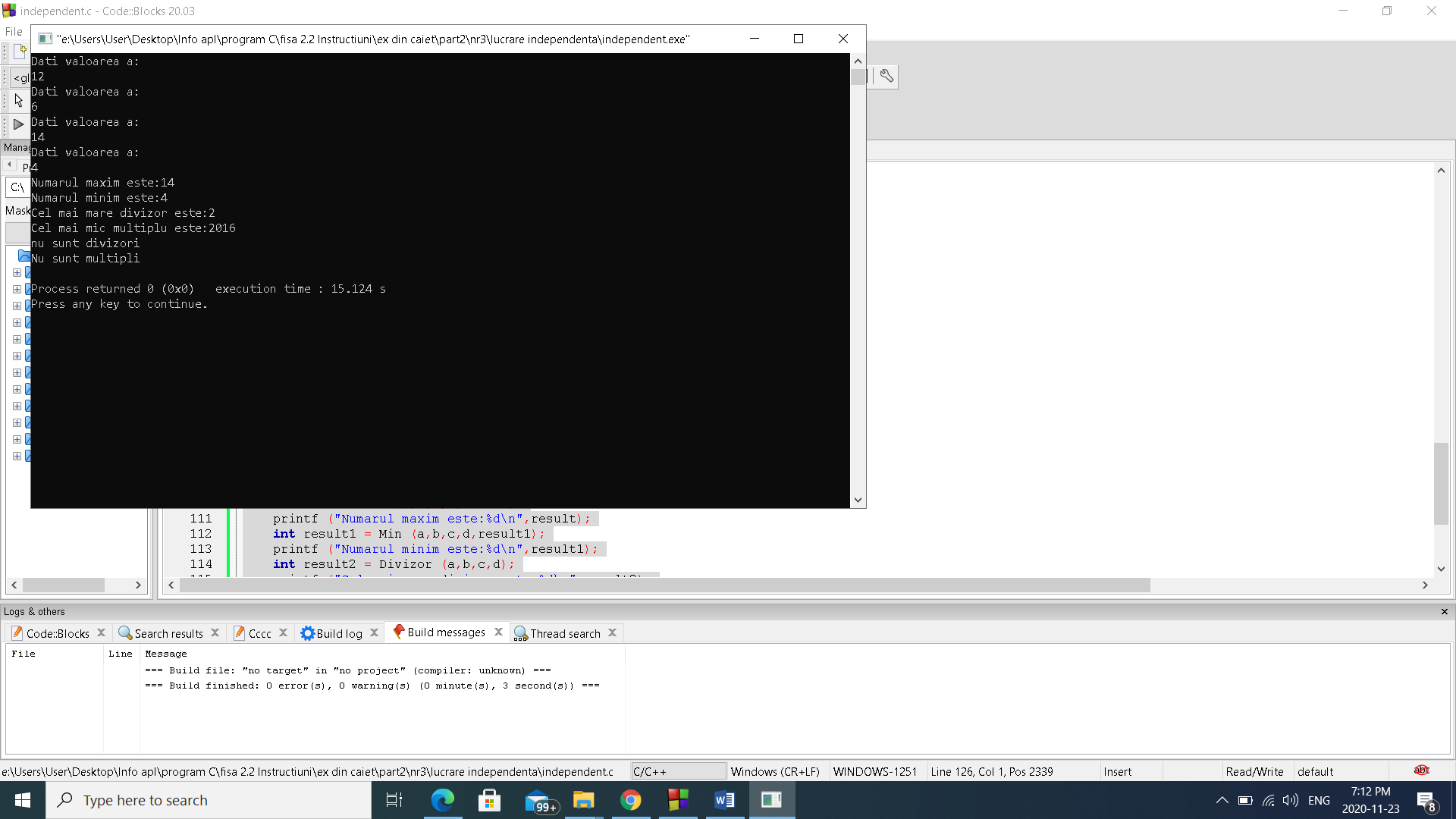
int result5 = verificare\_multiplu(a,b,c,d);

if (result5==0){

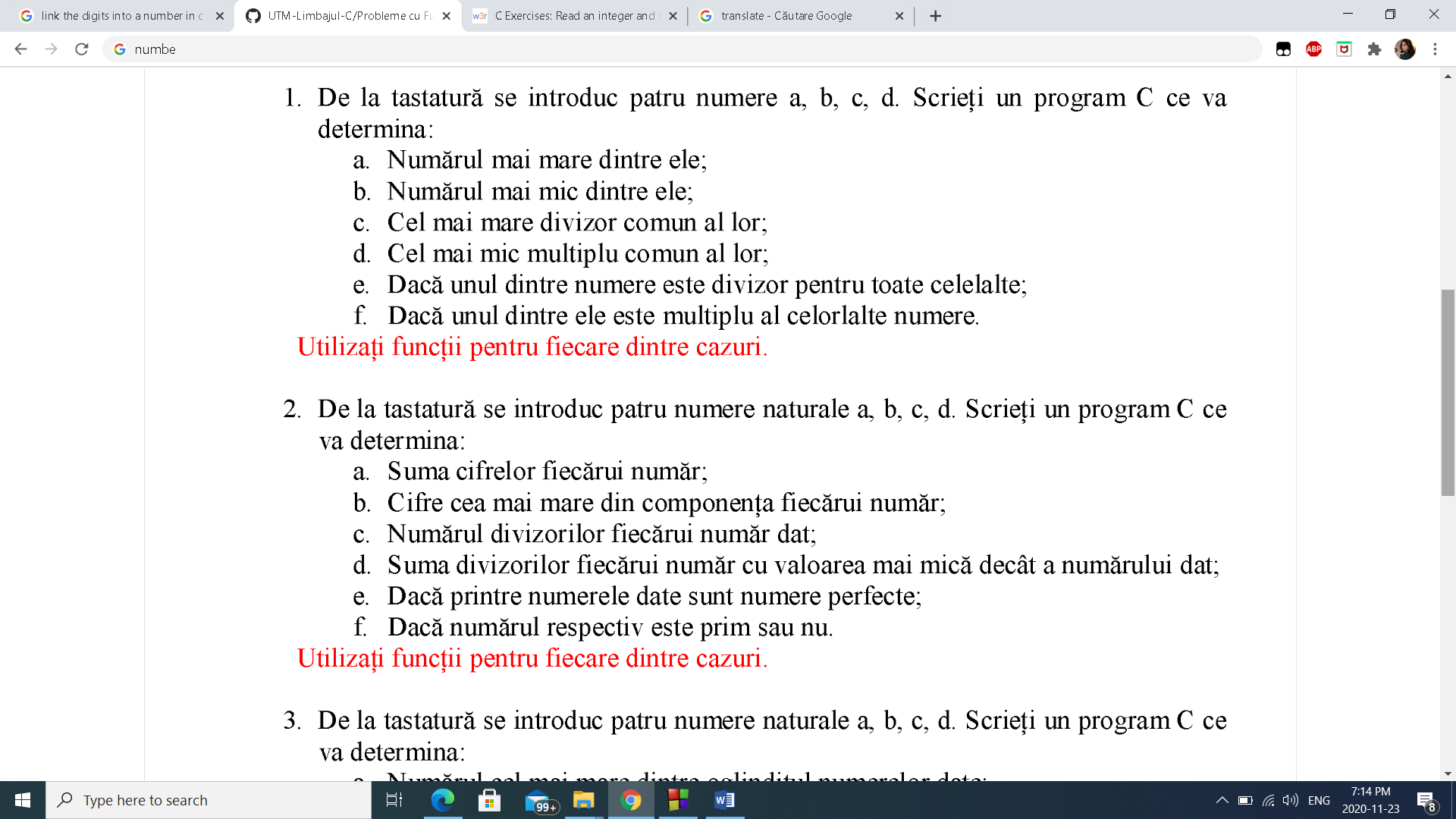
printf("Sunt multipli\n");}

else{printf("Nu sunt multipli\n");}}

Output:



Ex. 2



#include <stdio.h>

#include <stdlib.h>

int suma(int num)

{

int sum=0;

while(num!=0)

{ sum+=num % 10;

num = num / 10;}

return sum;

}

int largest\_digit( int n )

{

int remider, Largest= 0;

while (n > 0){

remider = n % 10;

if (Largest < remider){

Largest = remider;}

n = n / 10;}

return Largest;

}

int divizor(int x)

{

int i,count=0;

for(i = 1; i <= x; i++) {

if(x%i== 0)

count++;}

return count;}

int suma\_nr(int x)

{

int i,sum=0;

for(i = 1; i < x; i++) {

if(x%i== 0)

sum+=i;}

return sum;}

int number\_(int nr)

{

int i,sum=0 ;

for(i=1;i<nr;i++){

if(nr % i == 0)

sum=sum+i; }

if (sum == nr)

return nr;

else return 0;}

int nr\_prim(int prime)

{

int i,count=0;

for (i=1; i <=prime; ++i) {

if (prime % i == 0) {

count++;}}

if (count==2)

return 1;

else return 0;}

int main ()

{

int a,b,c,d;

printf("Dati valoarea a:\n",a);

scanf("%d",&a);

printf("Dati valoarea b:\n",b);

scanf("%d",&b);

printf("Dati valoarea c:\n",c);

scanf("%d",&c);

printf("Dati valoarea d:\n",d);

scanf("%d",&d);

int result = suma(a);

printf ("suma cifrei 'a' este:%d\n",result);

int resulta = suma(b);

printf ("suma cifrei 'b' este :%d\n",resulta);

int resultb = suma(c);

printf ("suma cifrei 'c' este:%d\n",resultb);

int resultc = suma(d);

printf ("suma cifrei 'd' este:%d\n\n",resultc);

int result1 = largest\_digit(a);

printf ("cea mai mare cifra a nr 'a' este:%d\n",result1);

int result2 = largest\_digit(b);

printf ("cea mai mare cifra a nr 'b' este:%d\n",result2);

int result3 = largest\_digit(c);

printf ("cea mai mare cifra a nr 'c' este:%d\n",result3);

int result4 = largest\_digit(d);

printf ("cea mai mare cifra a nr 'd' este:%d\n\n",result4);

int result5 = divizor(a);

printf ("nr divizorilor cifrei 'a' este :%d \n",result5);

int result6 = divizor(b);

printf ("nr divizorilor cifrei 'b' este :%d\n",result6);

int result7 = divizor(c);

printf ("nr divizorilor cifrei 'c' este :%d \n",result7);

int result8 = divizor(d);

printf ("nr divizorilor cifrei 'd' este :%d\n",result8);

int result9 = suma\_nr(a);

printf ("suma divizorilot este :%d\n",result9);

int result10 = suma\_nr(b);

printf ("suma divizorilot este :%d\n",result10);

int result11 = suma\_nr(c);

printf ("suma divizorilot este :%d\n",result11);

int result12 = suma\_nr(d);

printf("suma divizorilot este: %d\n",result12);

int result13 = number\_(a);

printf ("este nr perfect:%d\n",result13);

int result14 = number\_(b);

printf ("este nr perfect :%d\n",result14);

int result15 = number\_(c);

printf ("este nr perfect :%d\n",result15);

int result16 = number\_(d);

printf("este nr perfect: %d\n",result16);

int result17 = nr\_prim(a);

printf ("%d este nr prim \n",result17);

int result18 = nr\_prim(b);

printf ("%d este nr prim \n",result18);

int result19 = nr\_prim(c);

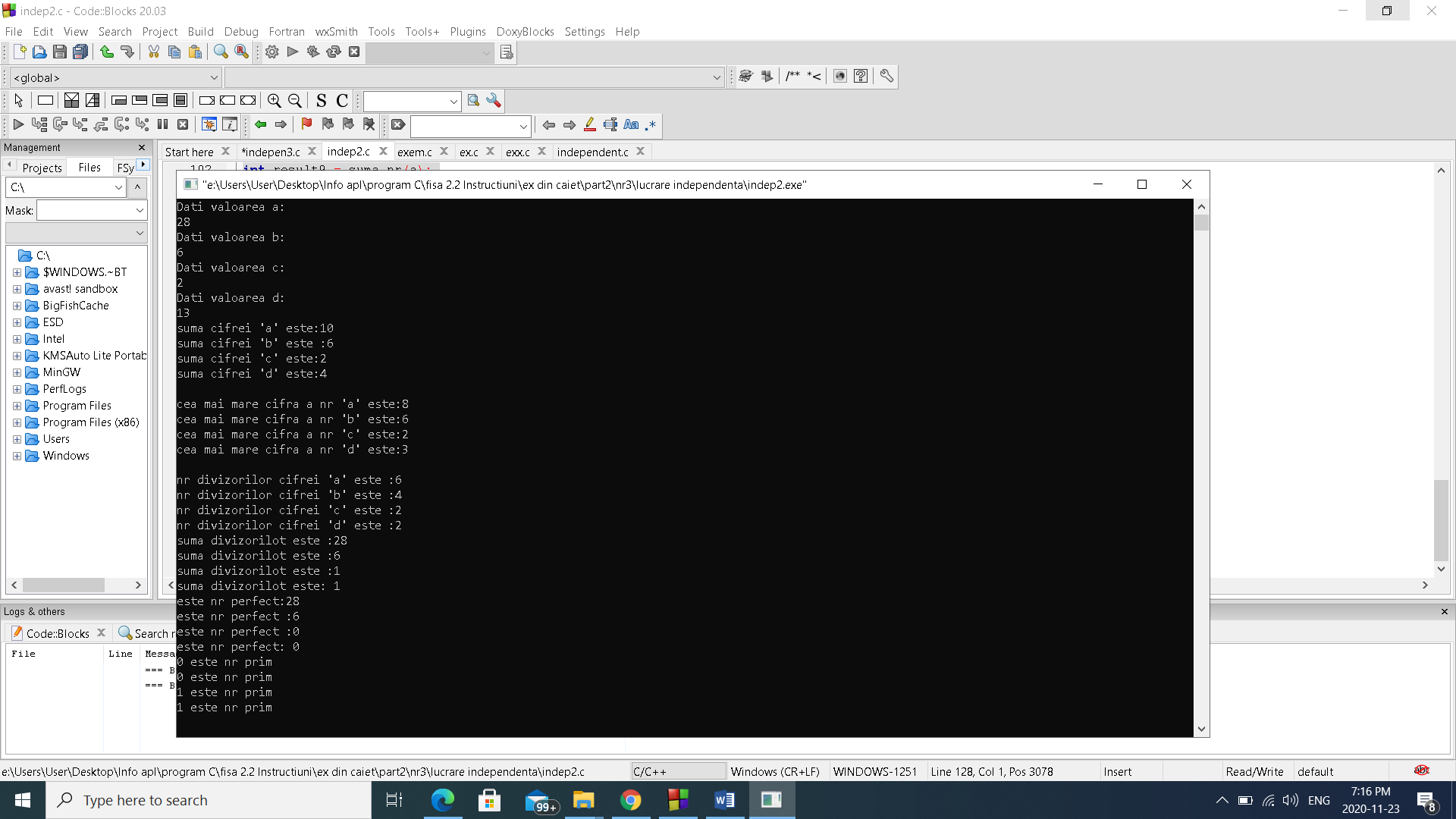
printf ("%d este nr prim \n",result19);

int result20 = nr\_prim(d);

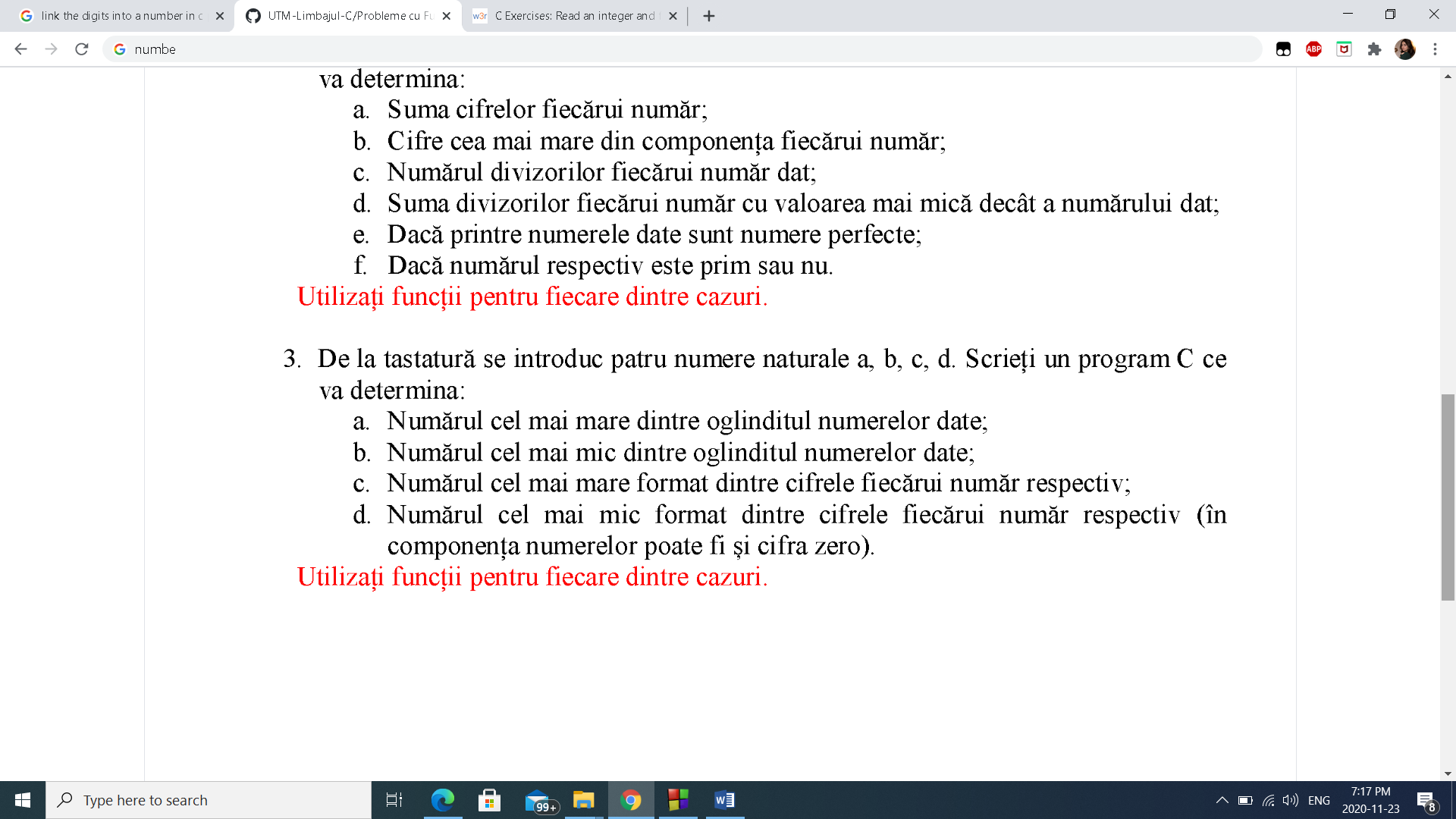
printf("%d este nr prim \n",result20);

}

Output:



Ex. 3



#include <stdio.h>

#include <stdlib.h>

int largest\_digit(int n)

{

int rev=0, rem, temp = 0, large = 0;

while (n != 0) {

rem= n%10;

rev= (rev\*10)+rem;

n/= 10;}

for(;rev>0; rev = rev/10){

temp = rev % 10;

if(temp > large){

large = temp;}}

return large;}

int smallest\_number (int n)

{

int rev=0, rem, temp = 0, large = 0,smallest;

while (n != 0) {

rem= n%10;

rev= (rev\*10)+rem;

n/= 10;}

smallest= rev%10;

while (rev > 0){

rem = rev % 10;

if (smallest> rem){

smallest = rem;}

rev = rev/10;}

return smallest;}

int nr\_format(int n)

{

int A[4],k=0, count=0;

do{

A[k]=n%10;

n=(n-A[k])/10;

k++;

count++;

}

while (n!=0);

int max;

for (int i=0;i<count-1;i++){

for (int k=i+1;k<count;k++){

if (A[i]<A[k]){

max=A[k];

A[k]=A[i];

A[i]=max;}}}

int nrf=0;

for (k=0;k<count;k++){

nrf=nrf\*10+A[k];}

return nrf;}

int nr\_for(int n)

{

int A[4],k=0, count=0;

do{

A[k]=n%10;

n=(n-A[k])/10;

k++;

count++;

}

while (n!=0);

int min;

for (int i=0;i<count-1;i++){

for (int k=i+1;k<count;k++){

if (A[i]>A[k]){

min=A[k];

A[k]=A[i];

A[i]=min;}}}

int nrf=0;

for (k=0;k<count;k++){

nrf=nrf\*10+A[k];

}return nrf;}

int main ()

{

int a,b,c,d;

printf("Dati valoarea a:\n",a);

scanf("%d",&a);

printf("Dati valoarea b:\n",b);

scanf("%d",&b);

printf("Dati valoarea c:\n",c);

scanf("%d",&c);

printf("Dati valoarea d:\n",d);

scanf("%d",&d);

int result = largest\_digit(a);

printf ("\ncifra cea mai mare a nr reversat 'a' este:%d\n",result);

int result1 = largest\_digit(b);

printf ("cifra cea mai mare a nr reversat 'b' este:%d\n",result1);

int result2 = largest\_digit(c);

printf ("cifra cea mai mare a nr reversat 'c' este:%d\n",result2);

int result3 = largest\_digit(d);

printf ("cifra cea mai mare a nr reversat 'd' este:%d\n\n",result3);

int result4 = smallest\_number(a);

printf ("cifra cea mai mica a nr reversat 'a' este:%d\n",result4);

int result5= smallest\_number(b);

printf ("cifra cea mai mica a nr reversat 'b' este:%d\n",result5);

int result6 = smallest\_number(c);

printf ("cifra cea mai mica a nr reversat 'c' este:%d\n",result6);

int result7 = smallest\_number(d);

printf ("cifra cea mai mica a nr reversat 'd' este:%d\n\n",result7);

int result8=nr\_format(a);

printf("nr nou format 'a' este: %d\n ",result8);

int result9=nr\_format(b);

printf("nr nou format 'b' este: %d\n ",result9);

int result10=nr\_format(c);

printf("nr nou format 'c' este: %d\n ",result10);

int result11=nr\_format(d);

printf("nr nou format 'd' este: %d\n\n ",result11);

int result12=nr\_for(a);

printf("nr nou format 'a' este: %d\n ",result12);

int result13=nr\_for(b);

printf("nr nou format 'b' este: %d\n ",result13);

int result14=nr\_for(c);

printf("nr nou format 'c' este: %d\n ",result14);

int result15=nr\_for(d);

printf("nr nou format 'd' este: %d\n\n ",result15);

}

Output:

