



Aplicații rezolvate

1. Se citește de la tastatură un număr natural n . Se cere să se calculeze suma divizorilor săi.

Exemplu: dacă $n=8$ divizorii săi sunt $\{1, 2, 4, 8\}$ deci va trebui să afișăm valoarea $15 = 1+2+4+8$

Algoritmul în pseudocod este:

```
citește n           //număr natural
s=0                 // inițializez suma divizorilor lui n cu zero
pentru d=1,n,+1    // caut posibilii divizori a lui n în intervalul [1,n]
┌
│   dacă n%d = 0 atunci //dacă d este divizor a lui n atunci
│   │   s=s+d           //il adaug la suma
│   └─┬─┘
└─┬─┘
   └─┘
scrie " suma divizorilor lui ",n," este ",s
```



```
#include<iostream>
using namespace std;
int n,s;
int main()
{
    cout<<"n="; cin>>n;
    for(int d=1;d<=n;d++)
        if(n%d==0)
            s=s+d;
    cout<<"Suma divizorilor lui"<<n<<" este "<<s;
    return 0;
}
```



2. Se citește de la tastatură un număr natural n . Se cere să se calculeze numărul divizorilor săi.

Exemplu: dacă $n=8$ divizorii săi sunt $\{1, 2, 4, 8\}$ deci va trebui să afișăm valoarea 4
Algoritmul în pseudocod este:

```
citește n           //număr natural
nr=0                // inițializez numărul divizorilor lui n cu zero
pentru d=1,n,+1    // caut posibili divizori a lui n în intervalul
[1,n]
|   dacă n%d = 0 atunci //dacă d este divizor a lui n atunci
|       nr=nr+1         //număr ca am gasit un divizor
|   sfârșit
sfârșit
scrie " numărul divizorilor lui ",n," este ",nr
```



```
#include<iostream>
using namespace std;
int n,nr;
int main()
{
    cout<<"n="; cin>>n;
    for(int d=1;d<=n;d++)
        if(n%d==0)
            nr++;
    cout<<"Numarul divizorilor lui"<<n;
    cout<<" este "<<nr;
    return 0;
}
```



3. Se citește de la tastatură un număr natural n . Se cere să se afișeze pe ecran numerele prime mai mici sau egale cu n .

Exemplu: dacă $n = 20$ atunci trebuie afișate valorile 2 3 5 7 11 13 17 19
Algoritmul în pseudocod este:



```
citește n //număr natural
dacă n<=1 atunci
    scrie "Nu exista numere prime mai mici sau egale cu 1"
altfel
    scrie "Numerele prime mai mici sau egale cu ", n, " sunt:"
    pentru x=2,n,+1 // consider fiecare numar x din intervalul [2,n]
        // verific daca numarul x este prim
        ok=1, d=2
        cât timp d<=radical din x și ok = 1 execută
            dacă x%d=0 atunci
                ok=0 // atunci x nu este prim
            altfel
                d=d+1
        dacă ok = 1 atunci // dacă x este prim atunci îl afișez
            scrie x, " "
```



```
#include<iostream>
#include<cmath>
using namespace std;
int n,s;
int main()
{ cout<<"n="; cin>>n;
  if (n<=1)
    cout<<"Nu exista numere prime mai mici sau egale cu 1!";
  else
    {cout<<"Numere prime mai mici sau egale cu "<<n<<" sunt :"<<endl;
      for(int x=2;x<=n;x++)
        { int d=2,ok=1;
          while (d<=sqrt(x) && ok)
            if (x%d==0)
              ok=0;
            else
              d++;
          if(ok)
            cout<<x<<" ";
        }
      }
  return 0;
}
```



4. Se dă un număr natural n . Se cere să se determine factorul la puterea cea mai mare care apare în descompunerea în factori primi a numărului dat.

Exemplu: dacă $n=24200$, atunci descompunerea în factori primi a lui este $24200=2^3 \cdot 5^2 \cdot 11^2$, se observă că factorul 2 apare la cea mai mare putere în descompunere, deci pe ecran trebuie afișată valoarea 2.
Algoritmul în pseudocod este:



```
citește n // număr natural
f_max=0 // factorul maxim initial este 0
p_max=0 // puterea acestui factor este initial tot 0
d=2 // primul posibil factor din descompunere este 2
cât timp n>=1 execută
    s=0 //număr de câte ori factorul d îl divide pe n
    cât timp n%d=0 execută
        s=s+1
        n=n/d
    dacă s ≠ 0 atunci //daca d a fost factor vad daca are cea mai mare putere
        dacă s > p_max atunci
            p_max=s; //actualizez puterea maxima
            f_max=d; //pastrez factorul la putrea maxima
        d
    d=d+1
d

scrie "Factorul care apare la puterea maxima in descompunerea in foactori
      primi este ", f_max
```



```
#include<iostream.h>
using namespace std;

int n,s,d,p_max,f_max;
int main()
{ cout<<"n="; cin>>n;
  d=2;
  while(n>1)
  { s=0;
    while(n%d==0)
    { s++;
      n=n/d;
    }
    if(s>p_max)
    { p_max=s;
      f_max=d;
    }
    d++;
  }
  cout<<"factorul la puterea maxima este
",
  cout<<f_max;
  return 0;
}
```



5. Se citesc de la tastatură n numere naturale. Se cere să se determine cel mai mare divizor comun dintre cele n numere.

Exemplu: dacă $n=4$ și numerele citite sunt 12, 48, 96 și 192 atunci cel mai mare divizor comun al lor este 12.

Pentru a determina acest lucru vom proceda astfel:

- Citim inițial n ca să stim câte numere vom prelucra
- Citim primele două numere
- Determinăm cmmdc dintre primele două numere citite
- Pentru celelalte $n-2$ numere procedăm astfel
 - Citim un număr
 - Determinăm cmmdc dintre numărul citit și cmmdc determinat anterior

Algoritmul în pseudocod este:



```
citește n // număr natural care precizează câte numere vom prelucra
//citește primele două numere și determină cmmdc dintre ele
citeste a,b
cât timp a ≠ b execută //dacă numerele sunt diferite atunci până când ajung
                        //egale din cel mai mare numar il scad pe cel mai mic
    dacă a > b atunci
        a = a-b
    altfel
        b = b-a
    ■
■
pentru i=3,n,+1
    citeste b //citeste următorul număr
    cât timp a ≠ b execută
        dacă a > b atunci
            a = a-b
        altfel
            b = b-a
        ■
    ■
■
scrie "cmmdc-ul numerelor citite este ", a
```



```
#include<iostream>
using namespace std;

int n,a,b;
int main()
{ cout<<"n="; cin>>n;
  cout<<"dati un numar:"; cin>>a;
  cout<<"dati un numar:"; cin>>b;
  while(a!=b)
    if(a>b)
      a=a-b;
    else
      b=b-a;
  for(int i=3;i<=n;i++)
    { cout<<"dati un numar:"; cin>>b;
      while(a!=b)
        if(a>b)
          a=a-b;
        else
          b=b-a;
    }
  cout<<"cmmdc dintre numerele date este ",a;
  return 0;
}
```