

Cuprins

Prelucrarea cifrelor lui n	2
Divizibilitate	8

Introducere

Urmărind cu atenție algoritmi prezentați, veți descoperii că există mici diferențe între ei, diferențe ce pot fi deduse simplu din citirea cu atenție a textului problemei. În plus, o bună stăpânire a acestor elemente de bază vor conduce la ușurință în aplicarea acestor tehnici și în cadrul altor algoritmi.

Prelucrarea cifrelor unui număr

Pseudocod	C++
<p>Algoritmul general</p> <pre> citeste n //initializari ┌daca n=0 atunci │ //initilizari │ ──┐ └─┘ ┌cat timp n≠0 executa │ uc←n%10 //extrag cifra │ n←[n/10] //tai cifra │ //prelucrez uc │ ──┐ └─┘ //afisari </pre>	<pre> #include <iostream> using namespace std; int main() { int n, uc; cin >> n; //initializari; if (n==0) //tratare caz n=0 { //initilizari } while (n!=0) { uc=n%10; //extrag cifra n=n/10; //tai cifra //prelucrez uc } //afisari; return 0; } </pre>
Numărări – câte cifre are n?	
<pre> citeste n k←0 ┌daca n=0 atunci │ k←1 │ ──┐ └─┘ ┌cat timp n≠0 executa │ uc←n%10 │ n←[n/10] │ k←k+1 │ ──┐ └─┘ scrie k </pre>	<pre> #include <iostream> using namespace std; int main() { int n, k, uc; cin >> n; k=0; if (n==0) { k=1; } while (n!=0) { uc=n%10; n=n/10; k=k+1; } cout << k; return 0; } </pre>

Numărări – câte cifre pare are n?	
<pre> citeste n k←0 ┌daca n=0 atunci │ k←1 │ ──┐ └───┘ ┌cat timp n≠0 executa │ uc←n%10 │ n←[n/10] │ ┌daca uc%2=0 atunci │ │ k←k+1 │ └───┘ └───┘ scrie k </pre>	<pre> #include <iostream> using namespace std; int main() { int n, k, uc; cin >> n; k=0; if (n==0) { k=1; } while (n!=0) { uc=n%10; n=n/10; if (uc%2==0) { k=k+1; } } cout << k; return 0; } </pre>
Numărări – are n cifre pare (da/nu) ?	
<pre> citeste n k←0 ┌daca n=0 atunci │ k←1 │ ──┐ └───┘ ┌cat timp n≠0 executa │ uc←n%10 │ n←[n/10] │ ┌daca uc%2=0 atunci │ │ k←k+1 │ └───┘ └───┘ ┌daca k=0 atunci │ scrie "nu" │ altfel │ scrie "da" └───┘ </pre>	<pre> #include <iostream> using namespace std; int main() { int n, k, uc; cin >> n; k=0; if (n==0) { k=1; } while (n!=0) { uc=n%10; n=n/10; if (uc%2==0) { k=k+1; } } if (k==0) { cout << "nu"; } else { cout << "da"; } return 0; } </pre>

Sume – suma cifrelor lui n	
<pre> citeste n k←0 ┌daca n=0 atunci │ k←1 │ ──┐ └─┬─┘ │ └cat timp n≠0 executa │ │ uc←n%10 │ n←[n/10] │ s←s+uc └─┬─┘ │ └scrie uc </pre>	<pre> #include <iostream> using namespace std; int main() { int n, s, uc; cin >> n; s=0; while (n!=0) { uc=n%10; n=n/10; s=s+uc; } cout << s; return 0; } </pre>
Sume – suma cifrelor impare ale lui n	
<pre> citeste n s←0 ┌cat timp n≠0 executa │ uc←n%10 │ n←[n/10] │ ┌daca uc%2=1 atunci │ │ s←s+uc │ └─┬─┘ │ │ └─┬─┘ │ └scrie s </pre>	<pre> #include <iostream> using namespace std; int main() { int n, s, uc; cin >> n; s=0; while (n!=0) { uc=n%10; n=n/10; if (uc%2==1) { s=s+uc; } } cout << s; return 0; } </pre>
Produs – produsul cifrelor lui n	
<pre> citeste n p←1 ┌daca n=0 atunci │ p←0 │ ──┐ └─┬─┘ │ └cat timp n≠0 executa │ │ uc←n%10 │ n←[n/10] │ p←p*uc └─┬─┘ │ └scrie uc </pre>	<pre> #include <iostream> using namespace std; int main() { int n, p, uc; cin >> n; p=1; if (n==0) p=0; while (n!=0) { uc=n%10; n=n/10; p=p*uc; } cout << p; return 0;} </pre>

Cifra **minimă**

```

citeste n
min←n%10
┌cat timp n≠0 executa
│   uc←n%10
│   n←[n/10]
│   ┌daca uc<min atunci
│   │   min←uc
│   └─┘
└─┘
scrie min
    
```

```

#include <iostream>
using namespace std;
int main()
{
    int n, min, uc;
    cin >> n;
    min=n%10;
    while ( n!=0 ) {
        uc=n%10;
        n=n/10;
        if ( uc<min ) {
            min=uc;
        }
    }
    cout << min;
    return 0;
}
    
```

Cifra **maximă** și de **câte** ori apare

```

citeste n
max←n%10
ap←1
n←[n/10]
┌cat timp n≠0 executa
│   uc←n%10
│   n←[n/10]
│   ┌daca uc>=max atunci
│   │   ┌daca uc>max atunci
│   │   │   max←uc
│   │   │   ap←2
│   │   │   altfel
│   │   │   ap←ap+1
│   │   └─┘
│   └─┘
└─┘
scrie max, " ",ap
    
```

```

#include <iostream>
using namespace std;
int main()
{
    int n, max, ap, uc;
    cin >> n;
    max=n%10;
    ap=1;
    n=n/10;
    while ( n!=0 ) {
        uc=n%10;
        n=n/10;
        if ( uc>=max ) {
            if ( uc>max ) {
                max=uc;
                ap=2;
            } else {
                ap=ap+1;
            }
        }
    }
    cout << max <<" " << ap;
    return 0;
}
    
```

Testul de palindrom și construcția inversului lui n	
<pre> citeste n aux←n inv←0 ┌cat timp n≠0 executa │ uc←n%10 │ n←[n/10] │ inv←inv*10+uc │ ──┘ └daca inv=aux atunci scrie "palindrom" altfel scrie "nu e palindrom" ──┘ </pre>	<pre> #include <iostream> using namespace std; int main() { int n, aux, inv, uc; cin >> n; aux=n; inv=0; while (n!=0) { uc=n%10; n=n/10; inv=inv*10+uc; } if (inv==aux) { cout << "palindrom"; } else { cout << "nu e palindrom"; } return 0; } </pre>
Eliminări de cifre din n – eliminarea cifrelor pare , construcții	
<pre> //in doi pasi citeste n inv←0 ┌cat timp n≠0 executa │ uc←n%10 │ n←[n/10] │ ┌daca uc%2=1 atunci │ │ inv←inv*10+uc │ │ ──┘ │ ──┘ └cat timp inv≠0 executa uc←inv%10 inv←[inv/10] n←n*10+uc ──┘ scrie n </pre>	<pre> //in doi pași #include <iostream> using namespace std; int main() { int n, inv, uc; cin >> n; inv=0; while (n!=0) { uc=n%10; n=n/10; if (uc%2==1) { inv=inv*10+uc; } } while (inv!=0) { uc=inv%10; inv=inv/10; n=n*10+uc; } cout<<n; return 0; } </pre>

Eliminări de cifre din n – **eliminarea cifrelor pare**, construcții

```
//direct
citeste n
nr←0
p←1
┌cat timp n≠0 executa
│   uc←n%10
│   n←[n/10]
│   ┌daca uc%2=1 atunci
│   │   nr←p*uc+nr
│   │   p←p*10
│   └─┘
└─┘
scrie nr
```

```
//direct
#include <iostream>
using namespace std;
int main()
{
    int n, nr, p, uc;
    cin >> n;
    nr=0;
    p=1;
    while ( n!=0 ) {
        uc=n%10;
        n=n/10;
        if ( uc%2==1 ) {
            nr=p*uc+nr;
            p=p*10;
        }
    }
    cout << nr;
    return 0;
}
```

Dublări de cifre ale lui n – **dublarea cifrelor de 5**

```
citeste n
nr←0
p←1
┌cat timp n≠0 executa
│   uc←n%10
│   n←[n/10]
│   nr←p*uc+nr
│   p←p*10
│   ┌daca uc%2=5 atunci
│   │   nr←p*uc+nr
│   │   p←p*10
│   └─┘
└─┘
scrie nr
```

```
#include <iostream>
using namespace std;
int main()
{
    int n, nr, p, uc;
    cin >> n;
    nr=0;
    p=1;
    while ( n!=0 ) {
        uc=n%10;
        n=n/10;
        nr=p*uc+nr;
        p=p*10;
        if ( uc%10==5 ) {
            nr=p*uc+nr;
            p=p*10;
        }
    }
    cout << nr;
    return 0;
}
```

Divizibilitate

General	
<pre> citeste n //initializari ┌ pentru d←1,n executa │ ┌ daca n%d=0 atunci │ │ //prelucrez d │ └─┘ └─┘ //afisari </pre>	<pre> #include <iostream> using namespace std; int main() { int n, d; cin >> n; //initializari; for (d=1;d<=n;d=d+1) { if (n%d==0) { //prelucrez d } } //afisari return 0; } </pre>
Numărări – câți divizori are n?	
<pre> citeste n k←0 ┌ pentru d←1,n executa │ ┌ daca n%d=0 atunci │ │ k←k+1 │ └─┘ └─┘ scrie k </pre>	<pre> #include <iostream> using namespace std; int main() { int n, k, d; cin >> n; k=0; for (d=1;d<=n;d=d+1) { if (n%d==0) { } } k=k+1; cout << k; return 0; } </pre>
Numărări – câți divizori pari are n?	
<pre> citeste n k←0 ┌ pentru d←1,n executa │ ┌ daca n%d=0 si d%2=0 atunci │ │ k←k+1 │ └─┘ └─┘ scrie k </pre>	<pre> #include <iostream> using namespace std; int main() { int n, k, d; cin >> n; k=0; for (d=1;d<=n;d=d+1) { if (n%d==0 && d%2==0) { k=k+1; } } cout << k; return 0;} </pre>

Testul de prim – prin numărare

```

citeste n
k←0
pentru d←1,n executa
┌─daca n%d=0 atunci
│   k←k+1
│   ──┘
└─┘
daca k=2 atunci
    scrie "prim"
altfel
    scrie "nu e prim"

```

```

#include <iostream>
using namespace std;
int main()
{
    int n, k, d;
    cin >> n;
    k=0;
    for ( d=1;d<=n;d=d+1 ) {
        if ( n%d==0 ) {
            k=k+1;
        }
    }
    if ( k==2 ) {
        cout << "prim";
    } else {
        cout << "nu e prim";
    }
    return 0;
}

```

Testul de prim – optimizat

```

citeste n
┌─daca n<1 atunci
│   ok←0
│   altfel
│   ┌─daca n=2 atunci
│   │   ok←1
│   │   altfel
│   │   ┌─daca n%2=0 atunci
│   │   │   ok←0
│   │   │   altfel
│   │   │   ok←1
│   │   │   d←3
│   │   │   ┌─cat timp d*d<n si ok=1 executa
│   │   │   │   ┌─daca n%d=0 atunci
│   │   │   │   │   ok←0
│   │   │   │   │   altfel
│   │   │   │   │   d←d+2
│   │   │   │   ──┘
│   │   │   ──┘
│   ──┘
└─┘
daca ok=1 atunci
    scrie "prim"
altfel
    scrie "nu e prim"

```

```

#include <iostream>
using namespace std;
int main()
{
    int n, ok, d;
    cin >> n;
    if ( n<1 ) {
        ok=0;
    } else {
        if ( n==2 ) {
            ok=1;
        } else {
            if ( n%2==0 ) {
                ok=0;
            } else {
                ok=1;
                d=3;
                while ( d*d<n && ok==1 ) {
                    if ( n%d==0 ) {
                        ok=0;
                    } else {
                        d=d+2;
                    }
                }
            }
        }
    }
    if ( ok==1 ) {
        cout << "prim";
    } else {
        cout << "nu e prim";
    }
    return 0;
}

```

Suma tuturor divizorilor lui n - simplu	
<p>citeste n s←0 ┌pentru d←1,n executa │┌daca n%d=0 atunci ││┌s←s+d ││└─┘ │└─┘ └─┘ scrie s</p>	<pre>#include <iostream> using namespace std; int main() { int n, s, d; cin >> n; s=0; for (d=1;d<=n;d=d+1) { if (n%d==0) { s=s+d; } } cout << s; return 0; }</pre>
Suma tuturor divizorilor lui n - optimizat	
<p>citeste n s←0 d←1 ┌cat timp d*d<=n executa │┌s=s+d ││┌daca d≠[n/d] atunci │││┌s←s+[n/d] │││└─┘ ││└─┘ │└─┘ └─┘ scrie s</p>	<pre>#include <iostream> using namespace std; int main() { int n, s, d; cin >> n; s=0; d=1; while (d*d<=n) { s=s+d; if (d!=n/d) { s=s+n/d; } d++; } cout << s; return 0; }</pre>

cmmdc	
<p>citeste a,b</p> <pre> ┌cat timp a≠b executa │ ┌daca a<b atunci │ │ a←a-b │ │ altfel │ │ b←b-a │ └─┘ └─┘ scrie a </pre>	<pre> #include <iostream> using namespace std; int main() { int a, b; cin >> a >> b; while (a!=b) { if (a<b) { a=a-b; } else { b=b-a; } } cout << a; return 0; } </pre>
cmmdc – cu resturi	
<p>citeste a,b</p> <pre> ┌cat timp b≠0 executa │ r←a % b │ a←b │ b←r │ └─┘ └─┘ scrie a </pre>	<pre> #include <iostream> using namespace std; int main() { int a, b, r; cin >> a >> b; while (b!=0) { r=a % b; a=b; b=r; } cout << a; return 0; } </pre>
cmmmc(a,b)=a*b/cmmdc(a,b)	
<p>citeste a,b</p> <pre> aa←a bb←b ┌cat timp b≠0 executa │ r←a % b │ a←b │ b←r │ └─┘ └─┘ cmmmc←aa*bb/a </pre>	<pre> #include <iostream> using namespace std; int main() { int a, b, aa, bb, r, cmmmc; cin >> a >> b; aa=a; bb=b; while (b!=0) { r=a % b; a=b; b=r; } cmmmc=aa*bb/a; return 0; } </pre>

