

Algoritmi elementari

1. Spargerea unui numar in cifre

```
if (n == 0) prelucrare_caz_special()
while (n > 0)
{
    cif = n%10;
    ..... //operatii care prelucreaza conform problemei cifra determinata
    n = n / 10;
}
```

a. calcularea numarului de cifre:

```
nrcifre = 0;
if (n == 0) nrcifre = 1;
while (n > 0)
{
    nrcifre++;
    n = n / 10;
}
```

b. determinarea primei cifre

```
if (n == 0) primacif = 0;
while (n > 9)
    n = n / 10;
primacif = n;
```

c. determinarea oglinditului (de ex., oglinditul numarului 12345 este 54321)

```
ogl = 0;
while (n > 0)
{
    cif = n % 10;
    ogl = ogl * 10 + cif;
    n = n / 10;
}
```

d. generarea cifrelor in ordinea in care apar in numar

```
p = 1;
while (p * 10 <= n)
    p = p * 10;
while (n > 0)
{
    cif = n / p;
    //prelucreaza cifra cif;
    n = n % p;
    p = p / 10;
}
```

e. numararea aparitiilor cifrei K

```
nrap = 0;
if (n == 0 && k == 0) nrap = 1;
while (n > 0)
{
    Cif = n % 10;
    if (cif == k) nrap++;
    n = n / 10;
}
```

f. eliminarea cifrelor pare

```
p = 1; nr = 0;
while (n > 0)
{
    Cif = n % 10;
    if (cif % 2 != 0)
    {
        nr = nr + cif * p;
        p = p * 10;
    }
    n = n / 10;
}
```

- verificarea lui **n** daca are **un numar impar de divizori (n este patrat perfect):**

```
sqrt(n) == int(sqrt(n))
```

- verificarea lui **n** daca are **3 divizori:**

```
sqrt(n) == int(sqrt(n)) si x=sqrt(n) numar prim (n este patrat perfect de numar prim)
```

2. Verificarea daca un numar este prim

```
if (n < 2) prim = 0;
else
{ prim = 1;          //presupunem ca n este prim
  for (d = 2; d * d <= n; d++)
    if (n % d == 0) {prim = 0; break;}
  if (prim == 1) .... //operatiile de efectuat cand n este prim
}
```

3. cmmdc(a,b)

```
rest = a % b;
while (rest != 0)
{
  a = b;
  b = rest;
  rest = a % b;}
cmmdc = b;
```

Atentie! Ca sa calculati **cel mai mic multiplu comun**, trebuie sa calculati cmmdc si sa aplicati formula **cmmmc=(ca*cb)/cmmdc;** unde ca=a; cb=b; copiile valorilor initiale, facute inainte de a calcula cmmdc

4. Determinarea divizorilor proprii ai lui n

```
for(d = 2; d <= n / 2; d++)
  if (n % d==0) ... //operatii care trebuie efectuate cu divizorul propriu d
                    // s = s + d; face suma divizorilor
                    // nrd++; numara divizorii proprii
```

5. Numarul de numere din intervalul [a,b], divizibile cu k (k>0)

Numarul de numere <=n divizibile cu k = n/k

Numarul de numere din [a,b], divizibile cu k = b/k - (a-1)/k

Numarul de multipli de a si multipli de b din [1,n] = n/cmmmc(a,b)

Numarul de multipli de a care **nu** sunt multipli de b din [1,n] = n/a-n/cmmmc(a,b)

6. Descompunerea in factori primi. Determinarea divizorilor primi ai lui n.

```
d = 2;
while (d * d <= n)
{
  m = 0;
  while ( n % d == 0)
    { m++; n = n / d;}
  if (m>0) ... //operatii care trebuie efectuate cu divizorul prim d la puterea m
  d++;
}
if (n>1) prelucreaza ultimul divizor prim n, care apare la puterea 1
```

7. Numarul divizorilor lui n. Formula lui Euler.

Formula lui Euler

Fie $N = p_1^{m_1} * p_2^{m_2} * \dots * p_k^{m_k}$

$N_{rdiv} = (m_1 + 1) * (m_2 + 1) * \dots * (m_k + 1)$

```
d = 2; nrdiv = 1;
while (d * d <= n)
{
  m = 0;
  while (n % d == 0)
    { m++; n = n / d;}
  nrdiv = nrdiv * (m + 1);
  d++;}
if (n>1) nrdiv = nrdiv * 2;
```

8. Cel mai mare numar fibonacci mai mic sau egal cu n

```
fin>>n;
f0 = f1 = 1;
while (f0+f1 <= n)
{
    f2 = f0 + f1;
    f0 = f1;
    f1 = f2;
}
Prelucreaza f1
```

9. Cifra de control a lui n

```
if (n == 0) cifc = 0;
else
    if (n % 9 == 0) cifc = 9;
    else cifc = n % 9;
```

10. Ultima cifra a lui x la puterea y (x, y numere naturale, x>0)

```
c = x % 10;
if (y % 4 == 1) ucif = c;
else if (y % 4 == 2) ucif = (c*c)%10;
    else if (y % 4 == 3) ucif = (c*c*c)%10;
        else if (y % 4 == 0) ucif = (c*c*c*c)%10;
```

11. Citirea pe rand a n numere

```
fin>>n;
for (i=1; i<=n; i++)
{
    fin>>a;
    .... //operatii care prelucreaza a cu oricare dintre ceilalti algoritmi
}
```

12. Citirea pe rand a n numere si prelucrarea perechilor de numere citite consecutiv (primul cu al doilea, al doilea cu al treilea, al treilea cu al patrulea, etc.)

```
fin>>n;
fin>>a;
for (i=2; i<=n; i++)
{
    fin>>b;
    .... //operatii care prelucreaza perechea formata din a si b
    a=b;
}
```

13. Citirea pe rand a n numere si determinarea minimului / maximului acestora

```
fin>>n;
fin>>a;
min=a; //presupunem ca minimul este primul element
for (i=2; i<=n; i++)
{
    fin>>a;
    if (min<a) min=a; //pentru maximul a n numere, in loc de (min<a) va fi (max>b)
}
```