

Algoritmi pentru prelucrarea cifrelor, cmmdc, divizori primi

Prelucrarea cifrelor unui număr

```
if (n == 0) //prelucrare numarul 0
while (n > 0) {
    cif = n%10;
    //operatii care prelucreaza
    //cifra curenta conform problemei
    n = n / 10;}
```

```
do{
    cif = n%10;
    //operatii care prelucreaza
    //cifra curenta conform problemei
    n = n / 10;
} while (n > 0);
```

ATENȚIE! Prin spargerea unui număr în cifre, valoarea inițială se distruge.
Dacă mai aveți nevoie de ea în program, trebuie să-i faceți o copie lui n înainte de a-l sparge

- a. calcularea numarului de cifre a lui n:

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

```
nrcif = 0;
do{
    nrcif++;
    n = n / 10;
} while (n > 0);
```

- b. determinarea primei cifre a numărului n

```
while (n > 9)
    n = n / 10;
prima_cifra = n;
```

- c. determinarea oglinditului lui n (numărul care conține cifrele lui n în ordine inversă)

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

ATENȚIE! La oglindirea unui număr terminat cu cifre de 0, aceste cifre 0 se pierd deoarece devin zerouri ne semnificative, în fața primei cifre a oglinditului.

- d. extragerea cifrelor lui n în ordinea în care apar în număr

```
//determinăm cea mai mare putere a lui 10, care este mai mică sau egală decât numărul n
p = 1;
while (p * 10 <= n) p = p * 10;
while (n > 0) {
    cif = n / p;
    //prelucreaza cif conform cerintelor problemei
    n = n % p;
}
```

e. Numărarea aparițiilor cifrei K în numărul n

```

nrap = 0;
do{
    cif = n % 10;
    if (cif == k) nrap++;
    n = n / 10;
} while (n>0);

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 cu o anumită proprietate din n (de ex. cifrele impare)

```

nrnou = 0;
p = 1;
do{
    cif = n % 10;
    if (cif % 2 == 0)
    {
        nrnou = nrnou + p * cif;
        p = p * 10;
    }
    n = n / 10;
} while (n>0);

```

Obs. Se construiește în variabila nrnou numărul care conține doar cifrele care NU sunt de eliminat, păstrând ordinea lor în numărul inițial.

Probleme propuse

1. Se citește un număr natural n de cel mult 15 cifre. Sa se afișeze elimine din n cifrele impare apoi sa se verifice daca numarul rezultat este palindrom (este egal cu oglinditul). Se va afisa numarul rezultat si valoarea 1 daca e palindrom sau 0 daca nu.
Ex: pentru n=245352102492, se vor afisa 2420242 1
2. Se citesc pe rand n numere naturale. Sa se afișeze numarul de cifre pare continute in toate aceste numere.
3. Se citesc pe rand n numere naturale nenule. Sa se afișeze sirul format cu prima cifra a fiecarui numar dat.
4. <http://pbinfo.ro>:

Divizori primi – vezi definiții și algoritm în Lectia 5 (fișa 55.pdf)

Probleme propuse

1. Se citește n un număr natural nenul. Sa se determine cel mai mic număr care are aceeași factori primi în descompunere ca și n.
Ex: pentru n=720, se va afisa 30 ($30=2 * 3 * 5$, $720=2^4 * 3^2 * 5$)
2. Se citesc 2 numere naturale nenule a, b ($1<a<b<1000$), sa se afișeze in ordine crescatoare numerele din intervalul [a,b] care au cel puțin 3 divizori primi. Daca nu exista astfel de valori, se va afisa 0.
3. Se citesc pe rand n numere naturale nenule, sa se afișeze numerele care au doar divizori primi la putere impara.
Ex. n=5 si numerele 24 16 9 17 28 se vor afisa 24 17 28

CMMDC, CMMC

Cel mai mare divizor comun (CMMDC) a două numere a și b

Algoritmul lui Euclid:

Pas 1. Se calculează restul împărțirii lui a la b.

Pas 2. Dacă rest este diferit de 0, a devine b, b devine rest iar restul se calculează împărțind a la b din nou și se reia Pas 2

Pas 3. Cmmdc este ultimul rest diferit de 0 din algoritm.

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

Cel mai mic multiplu comun (CMMC) a două numere a și b

$$\text{CMMC} = (a * b) / \text{CMMDC}$$

EX: $a = 36, b = 30, \text{cmmdc} = 6; \text{cmmc} = (36 * 30) / 6 = 180$

Doa numere **a si b** se numesc **prime între ele** dacă **CMMDC=1**

Ex: **25 si 44** sunt prime între ele pentru ca cel mai mare divizor comun este **1**

Probleme propuse

- Se citesc pe rand n numere naturale nenule. Sa se determine cmmdc pentru minimul si maximul din sir.
Ex. N=5 si valorile 12 8 9 28 17. Se va afisa 4 (cmmdc(8, 28)=4)
- Se citesc K un numar natural nenul apoi se citesc pe rand n numere naturale nenule. Sa se afiseze numerele prime cu K sau valoarea 0, daca nu exista astfel de numere.
Ex. K=15, N=5 si valorile 12 8 9 28 17. Se vor afisa 8 28 17
- Se citesc a si b doua numere naturale nenule reprezentand lungimile laturilor unei foi albe. Foia trebuie impartita in patratele de laturi egale, fara sa ramana resturi. Sa se determine latura maxima a unui patratel, numarul total de linii orizontale si verticale care trebuie trasate (pe marginile foii nu se traseaza linii) si numarul de patratele care se formeaza.
Ex: a=20, b=36. Se vor afisa valorile 4 12 45 (latura patratelului=4, numarul de linii pe latura a este 4 iar numarul de linii pe latura b este 8, deci 4+8=12, numarul total de patratele este 45)

Tema

<http://pbinfo.ro> problemele: [AfisareDivizoriComuni](#), [Divizorii Oglinditului](#), [Oglindit4](#), [Cmmdc2](#), [Pavare](#), [SumaCifreNrPrime](#), [SumCifPrim](#)

Trimiteți soluțiile pe adresa mirela.tibu@gmail.com sub forma unei arhive denumită cu numele vostru.

Creați arhiva urmând pașii:

- Creați un folder cu **numelevostru_tema6**
- Copiați una câte una sursele **main.cpp** în acest folder și redenumiți-le cu numele problemei
- Arhivați acest folder pastrand numele arhivei identic cu al folderului
- Atașați arhiva la email-ul pe care îl trimiteți la adresa mirela.tibu@gmail.com

Termen: 27 ianuarie 2016, ora 21

SUCCES!