Construirea unui număr cu cifre date

Să considerăm următorul șir de cifre, în ordine: 2 8 5 3

Cu ele se poate construi un număr, astfel:

* pornim de la valoarea R = 0;
* cifrele se adaugă în ordine:
  + la sfârșitul lui R
  + la începutul lui R

Dacă cifrele se adaugă la sfârșit, procedăm astfel:

1. R = 0
2. c = 2. R = 10 \* R + c, adică R devine 10 \* 0 + 2 = 2
3. c = 8. R = 10 \* R + c, adică R devine 10 \* 2 + 8 = 28
4. c = 5. R = 10 \* R + c, adică R devine 10 \* 28 + 5 = 285
5. c = 3. R = 10 \* R + c, adică R devine 10 \* 285 + 3 = 2853

Dacă cifrele se inserează la început, procedăm astfel:

1. R = 0
2. c = 2. R = R + 1 \* c, adică R devine 0 + 1 \* 2 = 2
3. c = 8. R = R + 10 \* c, adică R devine 2 + 8 \* 10 = 82
4. c = 5. R = R + 100 \* c, adică R devine 82 + 100 \* 5 = 582
5. c = 3. R = R + 1000 \* c, adică R devine 582 +1000 \* 3 = 3582

Ambele metode folosesc de fapt scrierea zecimală a numărului:

3582 = 0 + 1 \* 2 + 10 \* 8 + 100 \* 5 + 1000 \* 3

Pe de altă parte:

2853 =   
 285 \* 10 +3 =  
 (28\*10 + 5) \* 10 +3 =  
 ((2 \* 10 + 8)\*10 + 5) \* 10 +3 =  
 (((0 \* 10 + 2) \* 10 + 8)\*10 + 5) \* 10 +3

În practică, cifrele cu care se construiește numărul pot să provină din diverse surse. O situație frecventă este construirea unui număr folosind cifrele altui număr cunoscut.

Exemple

**Exemplul 1:** Determinarea oglinditului unui număr dat

Prin oglinditul (inversul) unui număr se înțelege un numărul scris cu cifrele numărului inițial, în ordine inversă. De exemplu, oglinditul lui 274 este 472, iar oglinditul lui 1300 este 31 – numerele nu pot să înceapă cu cifra 0.

**Rezolvare:**

* Fie n numărul dat, și ogl variabila în care vom calcula rezultatul.
* Inițial ogl = 0.
* Vom aplica procedeul de determinare prin trunchieri succesive a cifrelor lui n.
* Fiecare cifră a lui n, calculată prin n % 10 va fi adăugată la sfârșitul lui ogl, prin atribuirea ogl = 10 \* ogl + n % 10.

**Program C++:**

#include <iostream>

using namespace std;

int main(){

int n;

cin >> n;

int ogl= 0;

while(n){

ogl =10\*ogl + n%10;

n /= 10;

}

cout << ogl << endl;

return 0;

}

**Exemplul 2:** Se dă un număr natural. Să se modifice acest număr, micșorând cu o unitate fiecare cifră impară. Dacă numărul dat este 275 rezultatul va fi 264.

**Rezolvare:** Vom determina cifrele numărului dat și vom construi rezultatul, inserând cifrele la început. Cifrele pare se inserează ca atare, cifrele impare se inserează micșorate.

* Fie n numărul dat și R rezultatul. Vom utliliza o variabilă suplimentară, p, pentru a calcula puterile lui 10.
* Inițial R = 0, p = 1
* Vom determina prin trunchieri succesive cifrele lui n în variabila uc, uc = n % 10.
  + Dacă uc este par, R = R + p \* uc, apoi p = p \* 10.
  + Dacă uc este impar, R = R + p \* (uc - 1), apoi p = p \* 10.

**Program C++**

#include <iostream>

int main()

{

int n , R = 0, p = 1;

cin >> n;

while(n)

{

int uc = n % 10;

if(uc % 2 == 0)

R += p \* uc;

else

R += p \* (uc - 1);

p \*= 10;

n /= 10;

}

cout << R << std :: endl;

return 0;

}

Cifra de control a unui număr se obține efectuând suma cifrelor sale, apoi suma cifrelor acestei sume, până când suma obținută este un număr format dintr-o singură cifră. Această ultimă cifră poartă numele de cifră de control.

Exemplu: Cifra de control (cc)(��) a numărului n=4568248=4568248:  
cc(4568258)=cc(38)=cc(11)=2(4568258)=(38)=(11)=2.

Secvență de implementare C++:

while (n>9)

{

s=0;

while (n>0)

{

s+=n%10;

n/=10;

}

n=s;

}

Algoritmul prezentat este intuitiv, ușor de înțeles și contribuie la dezvoltarea gândirii algoritmice.

Tot pentru calcularea cifrei de control se poate folosi și următoarea metodă: se trunchiază numărul prin eliminarea ultimei cifre apoi se adună cu aceasta până pe obține un număr format dintr-o singură cifră.

Exemplu:

n=4568258=4568258  
n=456825+8=456833=456825+8=456833  
n=45683+3=45686=45683+3=45686  
n=4568+6=4574=4568+6=4574  
n=457+4=461=457+4=461  
n=46+1=47=46+1=47  
n=4+7=11=4+7=11  
n=1+1=2=1+1=2

Secvența de implementare C++:

while(n>9)

{

n=n/10+n%10;

}

În ambele cazuri timpul de execuție depinde de numărul de cifre ale numărului și de valorile acestora.

Pornind de la criteriul de divizibilitate cu 9, un număr se divide cu 9 dacă suma cifrelor sale este multiplu de 9 și cum cifra cea mai mare în baza 10 este 9, putem afla cifra de control calculând restul împărțirii numărului la 9. Dacă restul este 0 atunci cifra de control este 9, altfel este acel rest.

Secvența de implementare C++:

r=n%9;

if (r==0)

cc = 9;

else

cc = r;

În acest caz numărul de operații elementare nu mai depinde de valoarea lui n.