

Aplicații pentru scheme logice și pseudocod II

Prelucrarea cifrelor unui număr

- Problemele în care se cere prelucrarea cifrelor unui număr se rezolvă pornind de la câteva proprietăți matematice elementare;
- Prima dintre acestea este că prin împărțirea (cu rest) la **10** se izolează ultima cifră de celelalte cifre ale lui **n**;
- Mai precis, **$n \% 10$** (restul la împărțirea lui **n** la **10**) coincide cu ultima cifră a lui **n**, iar **$[n/10]$** (câtul la împărțirea cu **10** a lui **n**) corespunde numărului obținut din **n** prin eliminarea ultimei cifre;
De exemplu, $5832 \% 10 = 2$, iar $[5832/10] = 583$
- ATENȚIE! În limbajul C expresia corespunzătoare câtului la împărțirea cu **10** este **$n/10$** (fără paranteze pătrate).

Prelucrarea cifrelor unui număr

- În general, prin împărțirea cu rest la 10^k se izolează ultimele k cifre ale lui n de celelalte;

De exemplu, dacă $n = 5832$, $k = 3$ și $p = 10^3$, atunci $n \% p = 832$, iar $[n/p] = 5$

- În limbajul C nu există un operator care să poată fi folosit pentru ridicarea la putere a unui număr (există doar o funcție nerecomandată decât în puține cazuri);
 - Ridicarea la putere se poate face prin înmulțiri repetate.
-

Prelucrarea cifrelor unui număr

- Algoritmul general de prelucrare a cifrelor unui număr prelucrează și apoi elimină (prin împărțire la 10) ultima cifră a lui n , cât timp acesta e diferit de 0 (mai are cifre neprelucrate):
- Algoritmul general este prezentat mai jos în pseudocod și alăturat sub formă de schemă logică:

INIȚIALIZĂRI

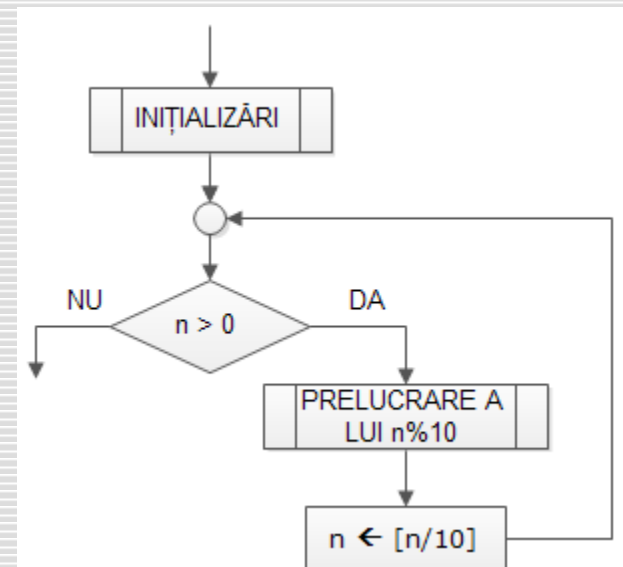
cât timp $n > 0$

{

 PRELUCREAZĂ $n \% 10$

$n \leftarrow [n/10]$

}



1) Numărul cifrelor lui n

- Scrieți un program care să rezolve următoarea problemă:
 - Se citește n (număr natural nenul, $n < 10000000000$);
 - Se cere să se afișeze numărul cifrelor lui n .
- De exemplu, dacă se citește $n = 64346$, se va afișa 5.
-

1) Numărul cifrelor lui n

- Algoritmul de rezolvare a problemei are în pseudocod forma de mai jos, iar schema logică se află în figura alăturată:

citește n

$nr \leftarrow 0$

cât timp $n > 0$

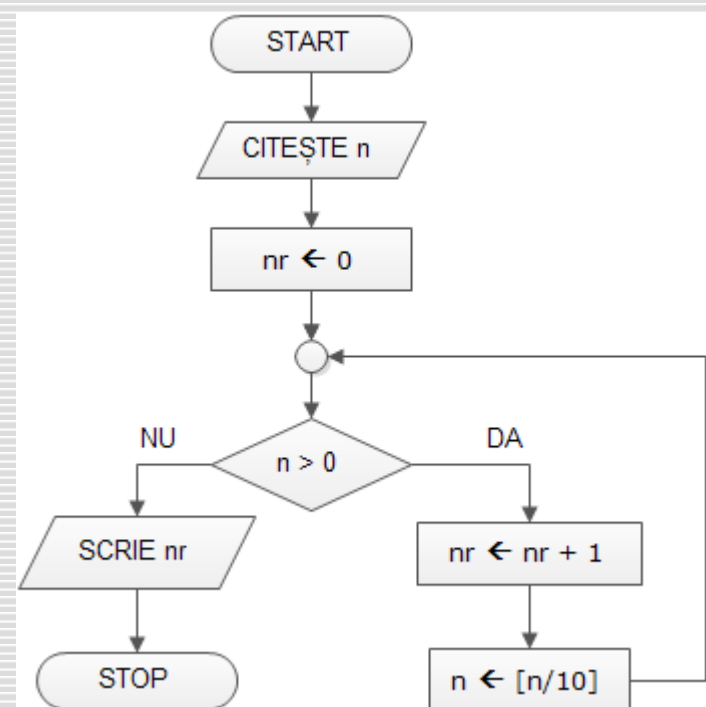
{

$nr \leftarrow nr + 1$

$n \leftarrow [n/10]$

}

scrie nr



1) Numărul cifrelor lui n

- ❑ Programul C se află în imaginea alăturată;
- ❑ Se folosesc variabile de tip **int**: numărul **n** citit este mai mic sau egal cu **100000000**, iar variabilele de tip **int** pot reține valori până la **$2^{31} - 1$** , care depășește **2000000000**.
- ❑ Pentru nr se poate folosi și un tip de date care ocupă mai puțină memorie, dar acesta urmează să fie studiat ceva mai târziu.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, nr;
6      scanf( "%d", &n );
7      nr = 0;
8      while( n > 0 )
9      {
10         nr++; // echivalent cu n = n + 1
11         n /= 10; //echivalent cu n = n / 10
12     }
13     printf( "%d", nr );
14     return 0;
15 }
16
```

2) Prima cifră a lui n

- Scrieți un program care să rezolve următoarea problemă:
 - Se citește n (număr natural nenul, $n < 1000000000$);
 - Se cere să se afișeze prima cifră a lui n .
- De exemplu, dacă se citește $n = 64346$, se va afișa 6.
-

2) Prima cifră a lui n

- Algoritmul de rezolvare a problemei are în pseudocod forma de mai jos, iar schema logică se află în figura alăturată:

citește n

cât timp $n > 0$

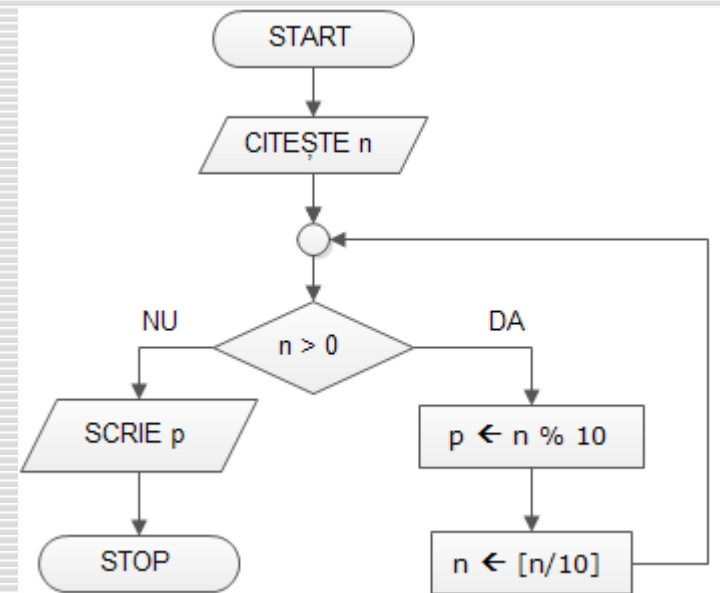
{

$p \leftarrow n \% 10$

$n \leftarrow [n/10]$

}

scrie p



2) Prima cifră a lui n

- Programul C alăturat rezolvă problema pe baza algoritmului descris în pseudocod și sub formă de schemă logică în slide-ul anterior;
- Principiul de funcționare e foarte asemănător cu cel al programului C pentru calculul numărului cifrelor lui **n**.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, p;
6      scanf( "%d", &n );
7      while( n > 0 )
8      {
9          p = n%10; // retin de fiecare data cifra pe care urmeaza sa o elimin
10         n /= 10; //echivalent cu n = n / 10
11     }
12     printf( "%d", p ); // ultima eliminata e prima cifra a valorii citite
13     return 0;
14 }
15
```

2) Prima cifră a lui n

- Se respectă tiparul general de rezolvare a problemelor care presupun prelucrarea cifrelor unui număr;
 - De data aceasta se folosește o variabilă p , în care e reținută de fiecare dată cifra care urmează să fie eliminată;
 - Cum cifrele sunt accesate de la dreapta la stânga, ultima prelucrată este de fapt prima cifră (de la stânga la dreapta) a valorii citite inițial.
-