

Lucrul cu cifrele unui număr

Atunci când se lucrează cu cifrele unui număr trebuie să fim atenți la cazul particular în care numărul citit este egal cu **0**. În acest caz, dacă se abordează rezolvarea cu ajutorul structurii repetitive *cât timp...execută*, atunci condiția ($n > 0$ sau $n \neq 0$) va fi falsă de la prima evaluare, nu se va intra în structura repetitivă și algoritmul se va termina fără a se face verificarea pentru $n=0$.

Exemplu: Se citește de la tastatură un număr natural cu cel mult 9 cifre. Se cere să se determine și să se afișeze câte cifre 0 conține numărul.

Dacă rezolvarea o realizăm ca mai jos, atunci pentru cazul în care $n=0$ valoarea afișată va fi **0**, deși ar fi trebuit să se afișeze **1**, pentru că are o cifră **0**.

Pseudocod	C++
<pre> citește n //se citește numărul nr←0 //variabila în care numărăm câte cifre 0 sunt în n cât timp n>0 execută // cât timp mai sunt cifre în număr uc←n%10 //determinăm ultima cifră dacă uc=0 atunci //dacă ultima cifră este 0 nr←nr+1 //numărăm cifra n←[n/10] //se elimină ultima cifră scrie nr //se afișează valoarea determinată </pre>	<pre> #include <iostream> using namespace std; int n,nr,uc; int main() { cin>>n; nr=0; while(n>0) { uc =n%10; if(uc==0) nr++; n=n/10; } cout<<nr; return 0; } </pre>

Pentru ca rezolvarea să fie corectă, adică să funcționeze și pentru cazul în care $n=0$ există două variante.

Varianta 1 – înainte de structura repetitivă se utilizează o structură alternativă în care se întreabă dacă $n=0$. În caz afirmativ se afișează 1, în caz negativ se utilizează rezolvarea anterioară.

Pseudocod	C++
<pre> citește n //se citește numărul dacă n=0 atunci scrie 1 altfel nr←0 //variabila în care numărăm câte cifre 0 sunt în n cât timp n>0 execută // cât timp mai sunt cifre în număr uc←n%10 //determinăm ultima cifră dacă uc=0 atunci //dacă ultima cifră este 0 nr←nr+1 //numărăm cifra n←[n/10] //se elimină ultima cifră scrie nr //se afișează valoarea determinată </pre>	<pre> #include <iostream> using namespace std; int n,nr,uc; int main() { cin>>n; if(n==0) cout<<1; else{ nr=0; while(n>0) { uc =n%10; if(uc==0) nr++; n=n/10; } cout<<nr;} return 0; } </pre>

Varianta 2 – în locul structurii retetitive *cât timp...execută* se folosește structura repetitivă *repetă...până când*. În cazul structurii repetitive *repetă...până când* se execută mai întâi propozițiile/instrucțiunile conținute, apoi se verifică expresia logică. Deci mai întâi execută conținutul, apoi verifică dacă $n=0$. Atenție la scrierea în C++!

Pseudocod	C++
<pre> citește n //se citește numărul nr←0 //variabila în care numărăm câte cifre 0 sunt în n repetă // cât timp mai sunt cifre în număr uc←n%10 //determinăm ultima cifră dacă uc=0 atunci //dacă ultima cifră este 0 nr←nr+1 //numărăm cifra n←[n/10] //se elimină ultima cifră până când n=0 scrie nr //se afișează valoarea determinată </pre>	<pre> #include <iostream> using namespace std; int n,nr,uc; int main() { cin>>n; nr=0; do { uc =n%10; if(uc==0) nr++; n=n/10; } while(n>0); cout<<nr; return 0; } </pre>

Exerciții:

1. Se citește de la tastatură un număr natural cu cel mult 9 cifre. Se cere să se determine și să se afișeze numărul cifrelor pare ale numărului.

Ex: dacă se citește numărul 32158, atunci se afișează 2; dacă se citește 0, atunci se afișează 1

2. Se citește de la tastatură un număr natural cu cel mult 9 cifre și o cifră nenulă k . Se cere să se determine și să se afișeze câte dintre cifrele numărului sunt strict mai mici decât k . Dacă nicio cifră din număr nu este strict mai mică decât k , atunci se afișează valoarea 0.

Ex: dacă se citește numărul 12593 și $k=7$, atunci se afișează 4; dacă se citește 0 și $k=3$, atunci se afișează 1