

Assignment 1

Filip Alina-Andreea, CR 1.2.A

April 22, 2019

1 Cerinta

Consideram o carte cu pagini numerotate. Numerotarea necesita un numar de n cifre. Proiecteaza un algoritm care sa determine numarul de pagini ale cartii. De exemplu, pentru o carte cu 500 de pagini, numarul de cifre utilizate este utilizare:

$$9(1\dots 9) + 2X90(10\dots 99) + 3X401(100\dots 500) = 9 + 180 + 1203 = 1392$$

Fiindu-ti dat numarul 1392(input) trebuie sa afisezi numarul 500(output).

Ni se cere sa construim un algoritm care sa determine numarul de pagini ce poate fi numerotate cu numarul de cifre puse la dispozitie. Input-ul este reprezentat de numarul de cifre, iar output-ul este reprezentat de numarul de pagini.

2 Algoritmi

Pentru realizarea acestui program am utilizat mai multi algoritmi impartiti in mai multe coduri sursa.

2.1 Algoritmul de calcul al numarului de pagini

NUMAR-PAGINI(n)

1. $r = n$;
2. $cif = 1$;
3. $nr = 9$;
4. *while*($r \geq cif$)
5. *if*($r \geq nr * cif$)
6. $pag = pag + nr$;
7. $r = r - cif * nr$;

8. $nr = nr * 10;$
9. $cif + 1;$
10. else
11. $nr = r / cif;$
12. $pag = pag + nr;$
13. break;

Acest algoritm calculeaza numarul de pagini ce pot fi obtinute din numarul total de cifre puse la dispozitie in input urmand urmatorii pasi:

- Intai algoritmul ii da lui r care reprezinta numarul ramas de cifre date valoarea introdusa de la tastatura n ;
- cif primeste valoarea 1 si reprezinta numarul de cifre folosite pentru a exprima numarul unei pagini
- nr reprezinta numarul de pagini care pot si reprezentate cu cif cifre
- Cat timp mai exista cifre ramase pentru a reprezenta macar inca o pagina cu cif cifre se executa secventa urmatoare
- De verifica daca numarul de cifre ramase este mai mare sau egal cu numarul de cifre necesar pentru a reprezenta numarul maxim de pagini cu cif cifre
- Daca da atunci:
- numarul maxim de pagini este adaugat la paginile cartii pag apeland SUMA
- se scade din cifrele ramase, cifrele folosite
- se inmulteste numarul maxim de pagini ce pot fi obtinute cu 10 deoarece pentru 1 cifra se pot forma maxim 9 pagini, 2...90, 3...900 si asa mai departe
- se incrementeaza cumarul de cifre folosite pentru a numerota o pagina cu o unitate
- Daca nu se indeplineste conditia atunci inseamna ca avem prea putine cifre ramase pentru a reprezenta numarul maxim de pagini ce ar putea fi obtinute si executam urmatoarea secventa:
- impartim numarul de cifre ramase pentru a vedea cate pagini mai putem forma (luam numai partea intreaga a impartirii deoarece nu putem numerota 4,5 pagini de exemplu, nu este posibil)
- adaugam paginile care mai pot fi formate la paginile cartii

- iesim din bucla while deoarece nu mai avem suficiente cifre ramase pentru a mai reprezenta pagini

In prima faza algoritmul avea si o suma pentru a aduna de fiecare data cifrele utilizate iar conditia de intrare in while era daca suma era mai mica sau egala cu numarul total de cifre insa am considerat ca era o variabila in plus pe care am putut sa o elimin cu cateva modificari.

In forma lui finala, algoritmul are o complexitate egala cu :

$$T(n) = 3 + n * [1 + 4m + (n - m) * 3]$$

$n-m$ poate fi considerat 1 deoarece atunci cand se intra pe else se face iesirea din bucla while. Asadar:

$$T(n) = 3 + n(1 + 4m + 3) = 3 + n + 4mn + 3n = 4n + 4mn + 3 = O(n)$$

Aici, n reprezinta numarul de iteratii ale while-ului. In cel mai bun caz, conditia din while nu este indeplinita niciodata , astfel vom avea o complexitate egala cu $O(1)$. In cel mai rau caz, se intra de $n-1$ ori si in while si in instructiunea if conducand la o complexitate egala cu $O(n^2)$.

2.2 Algoritmul de printare

PRINTEAZA(pag)

1. printf(pag);

Acest algoritm se apeleaza pentru a printa numarul de pagini ce pot fi formate pentru fiecare numar de cifre introdus. Acest algoritm are complexitatea egala cu 1. In functie de numarul de variabile introduse el este apelat de n ori.

3 Date experimentale

Datele experimentale au fost create cu ajutorul functiei random din main, sunt in numar de 10 si cuprind valori intre 0 si 100000. Un set de astfel de date este urmatorul: 2234, 19677, 8286, 7517, 6346, 1138, 18466, 18377, 3228, 18174;

3.1 Algoritmul de randomizare a variabilei introduse

Acest algoritm se gaseste in main

1. $m = 10$;
2. $srand((unsigned)time(t))$;
3. $for(i = 0; i < m; i++)$
4. $n = rand() * procent(i + 1) * 1000$

5. $pag = NUMAR - PAGINI(n)$

6. $PRINTEAZA(pag);$

Algoritmul de randomizare returneaza o valoare la intamplare in functie de momentul de timp la care este apelat. Functioneaza astfel:

- m reprezinta numarul de valori random care vor fi generate, in cazul asta 10
- $srand$ initializeaza functia $rand$ in momentul de timp t
- for pentru gestionarea apelurilor la functii pentru fiecare valoare random
- n primeste valoarea returnata de functia $rand$ care poate avea valori cuprinse intre 0 si $(i+1)*1000$
- pag primeste valoarea returnata de functia $NUMAR-PAGINI$
- apelarea functiei de printare pentru pag

3.2 Valori experimentale

Pentru 8286 cifre programul a executat astfel:

- am apelat functia $NUMAR-PAGINI$
- $in functie\ cif = 1, r = n (= 8286), nr = 9, pag = 0$
- $r \geq cif(8286 \geq 1)$ da, $decintra\ in\ while$
- $r > nr * cif(8286 > 9 * 1)$ da, intra in prima ramura a lui if
- $pag = pag + nr = 9$
- $r = r - nr * cif = 8286 - 9 = 8277$
- $nr = nr * 10 = 90$
- $cif + 1 = 2$
- verifica din nou conditia $r \geq cif(8277 \geq 2)$ da, intra iar in $while$
- $r > nr * cif(8277 > 90 * 2)$ da, intra pe prima ramura a lui if
- $pag = pag + nr = 9 + 90 = 99$
- $r = r - nr * cif = 8277 - 180 = 8097$
- $nr = nr * 10 = 900$
- $cif + 1 = 3$
- verifica din nou $r \geq cif(8097 \geq 3)$ da, intra in $while$

- $r > nr * cif(8097 > 2700)$ da, intra pe prima ramura a lui if
- $pag = pag + nr = 99 + 900 = 999$
- $r = r - nr * cif = 8097 - 2700 = 5397$
- $nr = nr * 10 = 9000$
- $cif + 1 = 4$
- verifica daca $r \geq cif(5397 \geq 4)$ da, intra in while
- $r > nr * cif(5397 > 36000)$ nu, intra pe a doua ramura a lui if
- $nr = r / cif = 5397 / 4 = 1349.25$ aici luam doar partea intreaga a impartirii deoarece nr este de tip int
- $pag = pag + nr = 999 + 1369 = 2348$
- iesim din bucla while
- returnam pag catre functia main
- apelam PRINTEAZA
- printam numarul de pagini le cartii 8286 -- > 2348

Pentru 0 cifre programul functioneaza astfel:

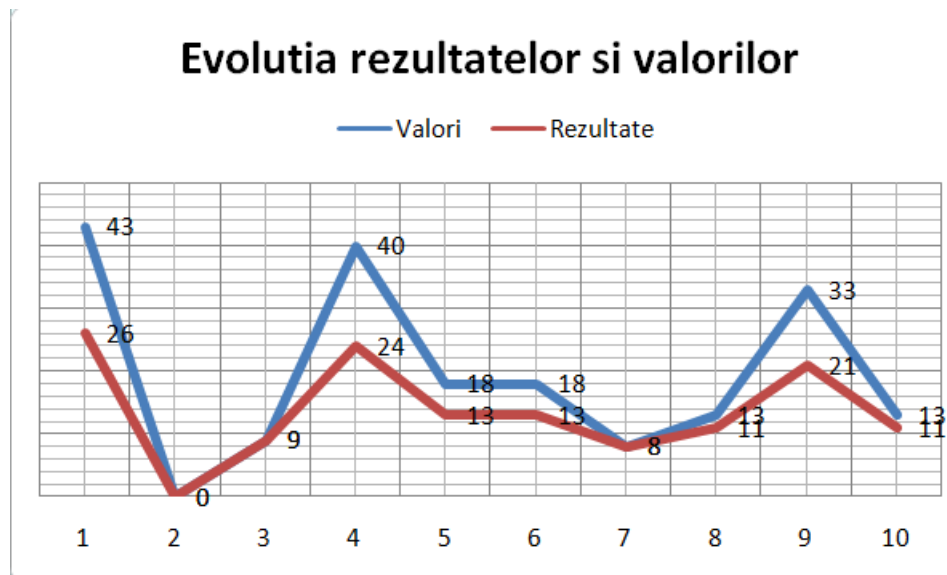
- am apelat functia NUMAR-PAGINI
- $infunctie\ cif = 1, r = n(= 0), nr = 9, pag = 0$
- $r \geq cif(0 \geq 1)$ nu, deci nu intra in while
- returnam pag catre functia main
- apelam PRINTEAZA
- printam numarul de pagini le cartii 0 -- > 0

Pentru 5 cifre programul functioneaza astfel:

- am apelat functia NUMAR-PAGINI
- $infunctie\ cif = 1, r = n(= 5), nr = 9, pag = 0$
- $r \geq cif(5 \geq 1)$ da, deci intra in while
- $r > nr * cif(5 > 9 * 1)$ nu, deci intra pe a doua ramura a lui if
- $nr = r / cif = 5 / 1 = 5$
- $pag = pag + nr = 0 + 5 = 5$

- iesim din while
- returnam pag catre functia main
- apelam PRINTEAZA
- printam numarul de pagini le cartii $5 - - > 5$

4 Rezultate si concluzii



In tabel se observa faptul ca graficul valorilor introduse se suprapune cu cel al rezultatelor pe distenta de la 0 la 9 deoarece pe distanta respectiva nu poti forma mai multe pagini decat numarul de cifre pus la dispozitie.

Ca si concluzie, am observat ca nu are rost sa apelam functia NUMAR-PAGINI decat pentru valori mai mari de 9, astfel incat am hotarat adaugarea a 2 if-uri in programul principal pentru a selecta modul de lucru al programului si pentru a sari peste pasii extra ce erau facuti in cazul lui 0 si 5. Astfel am obtinut un timp mai bun de executie al programului si o mai buna eficienta in cazul unor valori multiple in input.

Din acest proiect am invatat cum sa folosesc functia random, am invatat cum pot sa scap de variabilele in plus si cum sa fac un program sa functioneze mai eficient in diferite cazuri. De asemenea, am invatat cum sa lucrez cu fisiere sursa multiple si cum sa le integresc construind un program mult mai mare decat fiecare program existent in fisierele sursa.

Cea mai mare provocare a fost sa imi dau seama ce conditie sa pun in while si ce trebuie facut in cazul in care nu mai sunt destule cifre ramase pentru a reprezenta numarul maxim de pagini ce ar putea fi reprezentat. Pentru mine a fost destul de greu sa inteleg si de ce este mai usor sa lucrezi cu mai multe

fisiere sursa si cum sa realizez legaturile dintre ele. Pentru a intelege am vizionat tutoriale pe pagini web cat si pe youtube si am analizat codurile sursa puse la dispozitie la laboratoare.

Mi-a fost destul de greu sa lucrez si cu transmiterea prin referinta a variabilelor in functii deoarece nu am lucrat foarte mult cu ele. Am inteles cum sa fac asta tot din tutoriale , insa nu am reusit sa inteleg din codurile de la laborator.