

Problema 4

Student: Oprea Tudor
Grupa: 241

iunie 2023
București

I. Enunțul problemei

Fie numărul natural $n \geq 10$.

a) Să se elaboreze un program care să determine dacă numărul este palindrom; în caz afirmativ se cere să se tipărească numărul și un mesaj (de exemplu: Numarul 12421 este palindrom); în caz negativ se cere să se afișeze numărul și oglinditul său (de exemplu: Numarul 186335 este oglinditul numarului 533681).

b) Să se determine și să se afișeze numărul pozițiilor în care numărul n și oglinditul său coincid (adica: pe poziția respectivă în cele două numere apare aceeași cifră).

II. Breviar teoretic

Pentru a rezolva problema dată în limbajul C++, sunt necesare următoarele cunoștințe teoretice:

- Lucrul cu numere întregi: Programul va trebui să lucreze cu numărul natural n ca un număr întreg și să efectueze operații de inversare și verificare a palindromului.
- Folosirea lui `div(/)` și a lui `mod(%)`, pentru a extrage cifrele din număr:
 - vom utiliza `n%10`, adică restul împărțirii la 10 al numărului dat pentru a putea vizualiza concret ultima cifră a numărului.
 - vom folosi `n/10` pentru a împărți ca numere întregi numărul dat și 10, eliminând astfel cifra de la capătul lui n .
- Inversarea unui număr: Pentru a verifica dacă un număr este palindrom sau pentru a obține oglinditul unui număr, vom avea nevoie de abilitatea de a inversa un număr. Acest lucru poate fi realizat adăugarea succesivă la finalul unui număr nou a ultimei cifre din n , urmată de eliminarea acesteia din numărul inițial.
- Compararea și verificarea palindromului: Pentru a verifica dacă un număr este palindrom, va trebui să comparăm cifrele sale de la început și de la sfârșit și să verifici dacă sunt egale.
- Identificarea pozițiilor comune: Pentru a determina și a afișa pozițiile în care numărul n și oglinditul său coincid, vei trebui să parcurgem ambele numere în același timp și să identificăm pozițiile în care cifrele corespund.

Implementarea acestor funcționalități necesită cunoștințe de bază în programarea C++, inclusiv lucrul cu variabile, structuri de control (bucle și condiții), precum structura "cât timp(condiție)" ce execută respectivul bloc de instrucțiuni până când condiția devine falsă.

III. Soluții propuse

A. Soluția cu numere întregi

Soluția inițială constă într-un algoritm care citește numărul n și inițializează un oglindit cu zero. Îi facem o copie lui, pentru a nu îi altera valoarea în momentul în care se vor scoate cifrele, iar apoi, cât timp copia are cifre (adică este diferită de zero), se ia ultima cifră și se adaugă la finalul oglinditului, după care se taie de la finalul copieii.

Procesul se repetă până când se epuizează toate cifrele din copie, iar la final se verifică dacă numărul inițial n este egal cu oglinditul, caz în care n este palindrom, iar dacă nu, se afișează mesajul corespunzător.

Pentru punctul b, se parcurg tot cu while cifrele din n și din oglindit și se compară, în caz de egalitate crescând numărul pozițiilor identice.

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int n, copie, oglindit=0, numarPozitiiIdentice=0, cifra;
6
7      cout<<"n=";
8      cin>>n;
9
10     //Ii facem o copie lui n
11     copie=n;
12
13     while(copie!=0){
14
15         cifra = copie%10;
16         oglindit=oglindit*10 + cifra;
17         copie=copie/10;
18     }
19
20
21     //Daca numarul n (initial) are aceeași valoare cu oglinditul
22     if(n==oglindit)
23         cout<<"Numarul " <<n<<" este palindrom.";
24     else
25         cout<<"Numarul " <<oglindit<<" este oglinditul numarului " <<n<<" .";
26
27     cout<<endl;
```

```
28 //Pentru punctul b), verificam simultan
29 // daca ultima cifra din n este egala cu ultima din oglindit
30
31 while(n!=0){
32     cifra=n%10;
33     if(cifra==oglindit%10)
34         numarPozitiiIdentice++;
35
36     n=n/10;
37     oglindit=oglindit/10;
38 }
39 cout<<"Numarul citit si oglinditul sau au aceeasi cifra pe "<<numarPozitiiIdentice<<" pozitii.";
40
41
42 return 0;
43 }
```

B. Soluția cu șiruri de caractere

Soluția a doua presupune transformarea numerelor în siruri de caractere pentru a verifica dacă sunt sau nu palindrom, verificarea făcându-se cu doi indici care pleacă unul din stanga, iar altul din dreapta și avansează spre mijloc. La fiecare pas se compară elementul din prima jumătate cu cel din a doua, iar la găsirea unei incompatibilități se returnează false. Dacă funcția ajunge până la final și nu se găsesc două cifre diferite, se va returna true. Pentru punctul b) se parcurg cele doua stringuri iterativ de la 0 la len(n)-1 si se numara pozitiile identice.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 bool estePalindrom(int n) {
6     // Transformam numarul in sir de caractere
7     string nSir = to_string(n);
8
9     // Verificam daca sirul de caractere este palindrom
10    // comparand termeni din prima jumătate cu cei din a doua
11    int stanga = 0;
12    int dreapta = nSir.length() - 1;
13    while (left < right) {
14        if (nSir[stanga] != nSir[dreapta])
15            return false;
16        stanga++;
17        dreapta--;
18    }
19    return true;
20 }
21 }
```

```
22 int main()
23 {
24     int n, numarPozitiiIdentice=0;
25     cout << "n= ";
26     cin >> n;
27
28     if (estePalindrom(n))
29         cout << "Numarul " << n << " este palindrom.\n";
30     else{
31         cout << "Numarul " << n << " este oglinditul numarului ";
32         string nSir = to_string( val: n);
33         string oglindit( beg: nSir.rbegin(), end: nSir.rend());
34         cout<<oglindit<<" .\n";
35     }
36
37     string nSir = to_string( val: n);
38     string oglindit( beg: nSir.rbegin(), end: nSir.rend());
39     for (int i = 0; i < nSir.length(); i++)
40         if (nSir[i] == oglindit[i])
41             numarPozitiiIdentice++;
42
43     cout<<"Numarul citit si oglinditul sau au aceeasi cifra pe "<<numarPozitiiIdentice<<" pozitii.";
44
45
46     return 0;
47 }
```

Dintre cele două soluții, nu există una care să fie mai bună decât cealaltă, ambele făcând aproape aceiași pași spre găsirea rezultatului corect, diferit fiind doar modul de lucru - iterarea prin caracterele unui string și tăierea cifrelor unui număr. Nu se poate stabili dacă vreuna din cele două metode este cu adevărat mai eficientă, ambele dovedindu-se a fi complete și corecte pentru cerința propusă.

Complexitatea algoritmilor propuși este $O(n)$.

IV. Tratarea metodică

Enunțul problemei este complet și corect formulat. Este vorba despre verificarea dacă un număr este palindrom sau nu și determinarea pozițiilor în care numărul și oglinditul său au cifre comune.

Confuziile care pot apărea în enunț se referă la modul de interpretare a cerințelor, în special pentru partea b) a problemei. Uneori, termenul "poziții" poate fi interpretat în diferite moduri (poziții în cadrul șirului de caractere sau poziții numerice ale cifrelor). În acest context, "poziții" se referă la indexul cifrelor în șirul de caractere reprezentând numărul și oglinditul său. Deci se dorește numărarea acestor poziții pe care numerele sunt la fel atât în număr, cât și în oglindit.

O altă confuzie poate fi determinată de neînțelegerea exactă a cuvântului „oglindit”, adesea făcându-se confuzie între acesta (care reprezintă numărul cu cifrele în ordine inversă) și invers (care reprezintă $1/\text{numărul nostru}$, sau numărul la puterea -1).

Exemple de date de intrare:

- Pentru numărul $n = 12321$, programul va afișa "Numărul 12321 este palindrom."
- Pentru numărul $n = 186335$, programul va afișa "Numărul 186335 este oglinditul numărului 533681."
- Pentru numărul $n = 12345$, programul va afișa "Numărul 12345 este oglinditul numărului 54321."
-

Pentru a rezolva problema, se pot distinge următorii pași:

1. Se primește numărul n de la utilizator.
2. Se verifică dacă numărul este palindrom prin compararea cifrelor de la început și de la sfârșit în șirul de caractere reprezentând numărul.
3. Dacă numărul este palindrom, se afișează un mesaj corespunzător.
4. Dacă numărul nu este palindrom, se calculează oglinditul numărului și se afișează numărul și oglinditul.
5. Se găsesc și se afișează pozițiile în care numărul și oglinditul au cifre comune prin parcurgerea cifrelor în șirurile de caractere și compararea lor.

Problema se bazează pe conceptele de verificare a palindromului și parcurgerea cifrelor într-un șir de caractere. Poate fi generalizată pentru numere de orice lungime și se poate adapta și pentru alte baze numerice în loc de baza 10.

Cazurile particulare utile pot include numere care sunt palindroame, numere care nu sunt palindroame și numere cu cifre comune între număr și oglindit.

Greșeli tipice în rezolvarea problemei pot include: greșeli în calcularea oglinditului numărului, tratarea greșită a pozițiilor și indexării în șirul de caractere, sau omisiuni în verificarea palindromului.

O problemă similară ar putea fi formulată astfel: Să se scrie un program care primește un număr natural n și determină dacă acesta este palindrom sau nu, iar în caz afirmativ, afișează oglinditul lui n calculat exclusiv cu cifrele sale pare. Această problemă este similară, dar necesită calcularea încă unui oglindit, acesta din urmă având condiția de adăugare a cifrelor paritate a acestora. Calculul celor doi oglinditi se poate face simultan, printr-o singură parcurgere a cifrelor numărului n .

Această problemă poate fi utilizată într-un proiect didactic pentru a înțelege conceptele de parcurgere a șirurilor de caractere și manipularea numerelor în programare. De asemenea, poate fi utilizată pentru a dezvolta abilități de rezolvare a problemelor și gândire logică.

V. Barem de corectare

- | | |
|--|----------|
| • oficiu | 1 punct |
| • calculul oglinditului unui număr | 3 puncte |
| • verificare palindrom + mesaj corespunzător | 2 puncte |
| • calculul numărului de poziții identice | 2 puncte |
| • tipuri de date, declararea, citirea și inițializarea variabilelor | 1 punct |
| • corectitudine globală a programului - sintaxa, structura programului | 1 punct |

Orice soluție alternativă corectă primește punctaj maxim. Corectitudinea globală a programului include utilizarea corectă a subprogramelor, includerea bibliotecilor necesare lucrului, folosirea corectă a simbolurilor precum “; { }”.