

Problema B2 – Căutare binară

Accesarea informației digitale pentru volume foarte mari de date, necesită algoritmi eficienți de căutare care să permită localizarea informației utile cât mai rapid și folosind un minim de resurse de calcul. Un astfel de algoritm este algoritmul de căutare binară.

Cerință

Având la dispoziție un set de valori întregi, distincte, reprezentate sub forma unui vector, $V[]$, să se localizeze poziția unei anumite valori, x , specificată de utilizator, folosind un algoritm de căutare binară. Valorile din vector sunt mai întâi ordonate crescător. Dacă valoarea căutată x este identică cu valoarea din mijlocul vectorului, atunci algoritmul se încheie și valoarea a fost găsită. Dacă valoarea căutată este mai mică decât valoarea din mijlocul vectorului, procesul de căutare se repetă pentru sub-vectorul stâng, adică prima jumătate a valorilor din vector, comparând cu noua valoare din mijloc. Dacă valoarea căutată este mai mare decât valoarea din mijlocul vectorului, procesul de căutare se repetă pentru sub-vectorul drept, adică a doua jumătate a valorilor din vector și așa mai departe. Vectorul este divizat succesiv în două până când, fie este găsită valoarea căutată, fie sub-vectorul curent are dimensiunea unu. În cazul în care un sub-vector are un număr par de elemente, atunci se consideră valoarea din mijloc ca fiind implicit parte din acesta, ceea ce conduce mereu la un sub-vector cu un număr impar de elemente. Programul va afișa pe ecran toate valorile cu care este comparată valoarea x , adică valorile din mijloc până la finalizarea procesului de căutare.

Date de intrare

Se vor citi de la tastatură (fluxul *stdin*) următoarele date:

- o valoare întreagă pentru numărul de elemente din vector, n , urmată de caracterul *newline* (tasta *Enter*);
- valorile vectorului, $V[]$, introduse câte o valoare pe linie urmată de caracterul *newline* (tasta *Enter*);
- o valoare întreagă, x , ce reprezintă valoarea care va fi căutată, urmată de caracterul *newline* (tasta *Enter*).

Date de ieșire

Programul va afișa pe ecran la ieșire, toate valorile cu care este comparată valoarea x până la finalizarea algoritmului, câte o valoare pe linie, urmată de caracterul *newline* (tasta *Enter*).

ATENȚIE la respectarea cerinței problemei: afișarea rezultatelor trebuie făcută EXACT în modul în care a fost indicat! Cu alte cuvinte, pe stream-ul standard de ieșire nu se va afișa nimic în plus față de cerința problemei; ca urmare a evaluării automate, orice caracter suplimentar afișat, sau o afișare diferită de cea indicată, duc la un rezultat eronat și prin urmare la obținerea calificativului „Respins”.

Restricții și precizări

1. Dimensiunea vectorului $V[]$, n , este o valoare întreagă, pozitivă și impară, mai mare sau egală cu 7 și mai mică sau egală cu 21. Valorile vectorului sunt valori întregi, pozitive, în intervalul $[1;255]$. **Se asigură implicit faptul că valorile vectorului sunt diferite între ele;**
2. Valoarea căutată, x , este o valoare întreagă, pozitivă, în intervalul $[1;255]$;
3. **Atenție:** În funcție de limbajul de programare ales, fișierul ce conține codul trebuie să aibă una din extensiile *.c*, *.cpp*, *.java*, sau *.m*. Editorul web **nu va adăuga automat** aceste extensii și lipsa lor duce la imposibilitatea de compilare a programului!
4. **Atenție:** Fișierul sursă trebuie numit de candidat sub forma: $\langle \text{nume} \rangle . \langle \text{ext} \rangle$ unde *nume* este

numele de familie al candidatului și extensia este cea aleasă conform punctului anterior.
Atenție la restricțiile impuse de limbajul Java legate de numele clasei și numele fișierului!

Exemplu

Intrare	Ieșire
11 20 40 11 54 100 56 67 45 24 21 78 50	45 67 56 54
Explicație: $V=[20\ 40\ 11\ 54\ 100\ 56\ 67\ 45\ 24\ 21\ 78]$, $x=50$, $V_{\text{crescător}}=[11\ 20\ 21\ 24\ 40\ 45\ 54\ 56\ 67\ 78\ 100]$; Căutare: (1) $V_{\text{curent}}=[11\ 20\ 21\ 24\ 40\ 45\ 54\ 56\ 67\ 78\ 100]$, $\text{mijloc}=45$ (poziția 5), $50>45$, căutare în sub-vector dreapta. Valoare comparată=45; (2) $V_{\text{curent}}=[54\ 56\ 67\ 78\ 100]$, $\text{mijloc}=67$ (poziția 8), $50<67$, căutare în sub-vector stânga. Valoare comparată=67; (3) $V_{\text{curent}}=[54\ 56\ 67]$ (număr par de valori, se adaugă și centru), $\text{mijloc}=56$ (poziția 7), $50<56$, căutare în sub-vector stânga. Valoare comparată=56; (4) $V_{\text{curent}}=[54]$, $\text{mijloc}=54$, $50<54$, valoarea nu a fost găsită. Valoare comparată=54. Rezultat: 45 67 56 54.	
Intrare	Ieșire
11 20 40 11 54 100 56 67 45 24 21 78 78	45 67 78
Explicație: $V=[20\ 40\ 11\ 54\ 100\ 56\ 67\ 45\ 24\ 21\ 78]$, $x=78$, $V_{\text{crescător}}=[11\ 20\ 21\ 24\ 40\ 45\ 54\ 56\ 67\ 78\ 100]$. Căutare: (1) $V_{\text{curent}}=[11\ 20\ 21\ 24\ 40\ 45\ 54\ 56\ 67\ 78\ 100]$, $\text{mijloc}=45$ (poziția 5), $78>45$, căutare în sub-vector din dreapta. Valoare comparată=45; (2) $V_{\text{curent}}=[54\ 56\ 67\ 78\ 100]$, $\text{mijloc}=67$ (poziția 8), $78>67$, căutare în sub-vector din dreapta. Valoare comparată=67; (3) $V_{\text{curent}}=[67\ 78\ 100]$ (număr par de valori, se adaugă și centru), $\text{mijloc}=78$ (poziția 9), $78==78$, valoarea a fost găsită. Valoare comparată=78. Rezultat: 45 67 78.	

Timp de lucru: 120 de minute