

# Cerință Problemă

---

*Fie  $A$  un șir de  $n$  numere, cu  $n \geq 3$ . Să se scrie un program care să înlocuiască fiecare element nul din șir,  $a_k$ ,  $2 \leq k \leq n-1$ , cu:*  
*a) media geometrică a celui mai mare și celui mai mic element din șir;*  
*b) media geometrică a modului vecinilor săi direcți.*

---

## **EXPLICATII:**

Media geometrică a 2 numere este egală cu radicalul produsului celor 2. În cerință nu se specifică pe ce mulțime de numere lucrăm. Am ales să lucrăm pe  $\mathbb{Z}$ . În acest caz, trebuie să ținem cont de faptul că, sub radical, nu avem voie să folosim numere negative; deci produsul nu trebuie să fie negativ.

### **SUBPUNCTUL a)**

- Dacă minimul sau maximul este 0, atunci vectorul rămâne neschimbat.
- Dacă minimul este negativ și maximul pozitiv, nu putem calcula media geometrică.
- Dacă atât minimul cât și maximul sunt negative, atunci nu avem valori de 0, deci vectorul rămâne neschimbat.
- Dacă atât minimul cât și maximul sunt pozitive, mai mari decât 0, atunci nu avem valori de 0, deci vectorul rămâne neschimbat.
- Dacă minimul=maximul=0, atunci tot vectorul este constant, deci rămâne neschimbat.

**Concluzie:** Vectorul rămâne neschimbat, cu excepția să nu avem semne diferite la minim și maxim, caz în care vom afișa un avertisment.

### **SUBPUNCTUL b)**

Aici putem duce programul la bun sfârșit, pentru că lucrăm cu modulul. Singura mențiune pe care trebuie să o facem este faptul că dacă avem 0-uri pe poziții consecutive, atunci nu se modifică.

## **Breviar Teoretic:**

Subprogramele sunt părți esențiale ale unui program, care conțin tipuri de date, variabile și instrucțiuni destinate unei anumite prelucrări, identificabile prin nume, ce se pot activa la cerere prin intermediul acestor nume. Un program C++ este alcătuit dintr-o funcție principală, main, și mai multe funcții secundare. Aceste funcții secundare joacă un rol important, deoarece pot fi apelate din programul principal sau de către alte subprograme, făcând înțelegerea textului mai clară și venind în ajutorul programatorului pentru ca acesta să nu mai fie nevoit să rescrie aceleași linii de cod, de mai multe ori, în cadrul programului.

În cazul nostru, remarcăm faptul că afișarea vectorului este necesară pentru ambele subpuncte pentru a sublinia modificările aduse șirului inițial de numere.

Amintim cum se definește o funcție:

```
tip_returnat_de_funcție nume_funcție (listă_parametrii) // antetul
{
    Instrucțiuni; // corpul funcției
}
```

## COD REZOLVARE (C++):

```
1. #include <iostream>
2. #include <algorithm>
3. #include <vector>
4. #include <math.h>
5. using namespace std;
6.
7. //Afisare Vector
8. void afisare(vector<double> A)
9. {
10.     for(int i=0; i<A.size(); i++) cout<<A[i]<<" ";
11.     cout<<endl;
12. }
13.
14. int main()
15. {
16.     //Pas 1: Citire date de Intrare
17.     int n, i; //n = nr de elemente ale vectorului
18.     cout<<"Dati nr de elemente: "; cin>>n;
19.     if(n<3) {cout<<"Dati mai mult de 3 numere!"; return 0;}
20.
21.     double mini, maxi, aux;
22.     vector<double> A; //Sirul nostru
23.     A.resize(n); cout<<"Dati elementele sirului: ";
24.     for(i=0; i<n; i++) cin>>A[i]; //Citire Elemente ale Sirului A
25.
26.
27.     ///Subpunct a)
28.     //Pas 2: Identificarea minimului si maximului
29.     mini=min(A[0], A[1]);
30.     for(i=2; i<n; i++) mini=min(mini, A[i]);
31.     maxi=max(A[0], A[1]);
32.     for(i=2; i<n; i++) maxi=max(maxi, A[i]);
33.     //Pas 3: Verificare cazuri degenerate + Afisare
34.     cout<<"Subpunct a: "<<endl;
35.     if(mini*maxi<0) cout<<"Nu se poate calcula radicalul unui numar negativ."<<endl;
36.     else afisare(A);
37.
38.
39.     ///Subpunct b)
40.     for(i=1; i<n-1; i++)
41.         if(A[i]==0) A[i]=sqrt(abs(A[i-1]*A[i+1]));
42.     cout<<"Subpunct b: "<<endl;
43.     afisare(A);
44.
45.
46.     return 0;
47. }
```

## COMPLEXITATE: $O(n)$

## **Barem:**

- Din Oficiu 1p
- Cunoștințe Generale Necesare 1p
- Răspunsuri Preliminare 2p
- Lucrul cu Vectori (bonus 1p dacă se folosește biblioteca „vector”) 1p
- Scrierea unei Funcții de Afișare (pentru a nu repeta codul) 1p
- Rezolvare propriu-zisă 3p
  - Lucrul cu biblioteca math.h 1p
  - Lucrul cu biblioteca algorithm 1p
  - Funcționabilitate și cod 1p
- Stil (comentarii, indentare) 1p

**Observație:** Se acordă punctaje parțiale. Spre exemplu, dacă elevul nu folosește funcția „min” implementată în biblioteca „algorithm” și o scrie de mână, va primi 0.5p.