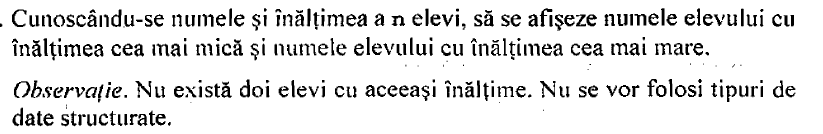
Corectați și completați următorul enunț, apoi propuneți o rezolvare, respectând etapele rezolvării unei probleme de informatică:



**Elemente care fac enunțul incomplet/incorect:**

* Nu spune cum sunt citite numele și înălțimile (de la tastatură sau sunt stocate deja).
* Nu se precizează clar dacă se folosesc vectori sau alte structuri.
* Nu oferă un exemplu de input/output ca să fie mai ușor de înțeles.
* Nu se precizează ce se întâmplă în caz de introducere a unor date invalide (ex: text în loc de număr, înălțimi negative etc.).

**Enunțul complet și corect:**

Se citesc de la tastatură n elevi, fiecare având un **nume** și o **înălțime** (valoare întreagă pozitivă). Să se afișeze:

* numele elevului cu înălțimea cea mai mică
* numele elevului cu înălțimea cea mai mare

Observații:

* Nu există doi elevi cu aceeași înălțime.
* Nu se folosesc tipuri de date structurate (se folosesc două liste: una pentru nume și una pentru înălțimi).
* Dacă se introduc date invalide (de exemplu, un text în loc de număr, sau o înălțime negativă), programul trebuie să afișeze un mesaj de eroare și să se oprească.

**Exemplul:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| n = 3  Ana 160  Mihai 175  Elena 155 | Minim: Elena  Maxim: Mihai |

**Analiza exemplului:**

* Se citesc listele: cu nume, cu înălțimi.
* Parcurgem lista cu înălțimi și salvăm pozițiile unde avem cel mai mic și cel mai mare număr.
* Afișăm numele de pe acele poziții din lista cu nume.

**Proiectarea algoritmului:**

1. Se citește n.
2. Se citesc n perechi: nume + înălțime, stocate în doi vectori.
3. Se inițializează min\_pos și max\_pos cu 0.
4. Se parcurg înălțimile și se actualizează pozițiile pentru minim și maxim.
5. Se afișează nume[min\_pos] și nume[max\_pos].

**Implementarea în limbaj de programare:**

#include <iostream>

#include <string>

using namespace std;

int main() {

int n;

cout << "n = ";

cin >> n;

if (n <= 0 || n > 100) {

cout << "Eroare: n trebuie sa fie intre 1 si 100." << endl;

return 1;

}

string nume[100];

int inaltimi[100];

for (int i = 0; i < n; i++) {

cout << "Nume: ";

cin >> nume[i];

cout << "Inaltime: ";

cin >> inaltimi[i];

if (inaltimi[i] <= 0) {

cout << "Eroare: inaltimea trebuie sa fie un numar pozitiv!" << endl;

return 1;

}

}

int min\_pos = 0, max\_pos = 0;

for (int i = 1; i < n; i++) {

if (inaltimi[i] < inaltimi[min\_pos]) {

min\_pos = i;

}

if (inaltimi[i] > inaltimi[max\_pos]) {

max\_pos = i;

}

}

cout << "Minim: " << nume[min\_pos] << endl;

cout << "Maxim: " << nume[max\_pos] << endl;

return 0;

}

**Testarea:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| n = 3  Alex 180  Daria 170  Vlad 190 | Minim: Daria  Maxim: Vlad |

|  |  |
| --- | --- |
| **Input** | **Output** |
| n = 2  Ioana 165  George 172 | Minim: Ioana  Maxim: George |

|  |  |
| --- | --- |
| **Input** | **Output** |
| n = 2  Andrei -170  Mara 165 | Eroare: inaltimea trebuie sa fie un numar pozitiv! |