

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

. Cunosându-se numele și înălțimea a n elevi, să se afișeze numele elevului cu înălțimea cea mai mică și numele elevului cu înălțimea cea mai mare.

Observație. Nu există doi elevi cu aceeași înălțime. Nu se vor folosi tipuri de date structurate.

Elemente care fac enunțul incomplet/incorect:

- tipul datelor lipsește (ce reprezintă n , ce liste vom avea, ce tip de dată e numele, înălțimea)
- limitele lipsesc (nr. Minim și maxim de elevi)
- cum se face afișarea?
- nu specifică nimic de vreo listă de elevi, cum va fi citită și memorată (daca nu folosește struct, atunci va trebui să specifice că dorește două liste, una cu numele (prenume/nume de fam/tot numele) și una cu înălțimea unde poziția e elementul comun dintre cele două liste ca să vorbim de același elev)
- tipuri de date “struct”, date structurate pot fi multe :)
- situațiile speciale nu sunt abordate
- în viața reală, vor exista doi elevi cu aceeași înălțime așa că ideal e să acceptăm ce e în realitate

Enunțul complet și corect:

Citire:

Se citește un număr natural n de la tastatură reprezentând numărul de elevi al unei clase. Se citesc în două liste, lista de cuvinte „nume” care va conține prenumele elevilor și lista de numere reale „înălțime”. Pentru fiecare elev se va citi prenumele (doar unul, caractere legate fără spații) și înălțimea sa, corespundente la același index.

Afișare:

Pe prima linie numele celui mai scund elev, pe a doua linie numele celui mai înalt elev. Dacă apar doi elevi cu aceeași înălțime afișați mesajul „Elevii au aceeași înălțime.”. Dacă nu sunt destui elevi, afișați mesajul: „Există doar un elev.”

Condiții:

$1 \leq n < 100$

Un nume poate avea maxmi 30 de caractere, fara spatii.

A nu se folosi noi tipuri de date înafară de cele existente în C++.

Exemplul:

Date de intrare:

5
Andrei 1,50
Maria 1,60
Mihai 1,70
Ioana 1,45
Dan 1,90

Date de ieșire:

Ioana
Dan

Analiza exemplului (termeni naturali):

Avem 5 elevi, deci automat știm că nu avem cazul cu un singur elev.

Parcurgem elementele rând pe rând.

Andrei are 1,50. Momentan el e și cel mai înalt și cel mai scund pentru că nu avem termeni de comparație așa că vom ține minte această informație.

Trecem mai departe la Maria. Maria are 1,60. 1,60 este mai mare decât 1,50 al lui Andrei însemnând că Andrei e cel mai scund și Maria cea mai înaltă de până acum.

Mihai are 1,70. E mai înalt decât Maria, deci acum Mihai e cel mai înalt, iar Andrei rămâne cel mai scund.

Ioana are 1,45, mai mică decât Andrei care are 1,50, astfel, acum cel mai scund elev este Ioana, iar Mihai rămâne în continuare cel mai înalt.

Ajungem la Dan, iar el are 1,90, număr mai mare decât 1,70 însemnând că Dan este mai înalt decât Mihai, astfel ajungem la faptul că Dan e cel mai înalt elev, iar Ioana cel mai scund elev.

Lista se termină, astfel că acum suntem la final, știm că au înălțimi diferite, afișăm Ioana și Dan.

Analiza exemplului (termeni informatici):

date de intrare:

$n = 5$

Primul element:

- se verifica daca $n = 1$, $n = 5 \Rightarrow$ continuam

- se citeste $\text{nume}[1] = \text{Andrei}$ si $\text{inaltime}[1] = 1.5$

- se initializeaza $\text{maxim} = \text{inaltime}[1](1.5)$, $\text{minim} = \text{inaltime}[1](1.5)$, $\text{poz_minim} = 1$, $\text{poz_maxim} = 1$

Al doilea element $I = 2$:

- se citeste $\text{nume}[2] = \text{maria}$ si $\text{inaltime}[2] = 1.6$

- se verifica daca $\text{inaltime}[2] < \text{minim}$:

 - $1.6 < 1.5$ fals

- se verifica daca $\text{inaltime}[2] > \text{maxim}$:

 - $1.6 > 1.5$ adevarat \Rightarrow acum $\text{maxim} = 1.6$ si $\text{poz_maxim} = 2$

Al treilea element $I = 3$:

- se citeste $\text{nume}[3] = \text{Mihai}$ si $\text{inaltime}[3] = 1.7$

- se verifica daca $\text{inaltime}[3] < \text{minim}$:

 - $1.7 < 1.5$ fals

- se verifica daca $\text{inaltime}[3] > \text{maxim}$:

 - $1.7 > 1.6$ adevarat \Rightarrow acum $\text{maxim} = 1.7$ si $\text{poz_maxim} = 3$

Al patrulea element $I = 4$:

- se citeste $\text{nume}[4] = \text{Ioana}$ si $\text{inaltime}[4] = 1.45$

- se verifica daca $\text{inaltime}[4] < \text{minim}$:

 - $1.45 < 1.5$ adevarat \Rightarrow acum $\text{minim} = 1.45$ si $\text{poz_minim} = 4$

- se verifica daca $\text{inaltime}[4] > \text{maxim}$:

 - $1.6 > 1.5$ fals

Al 5-lea element $I = 5$:

- se citeste $\text{nume}[5] = \text{Dan}$ si $\text{inaltime}[5] = 1.9$

- se verifica daca $\text{inaltime}[5] < \text{minim}$:

 - $1.9 < 1.45$ fals

- se verifica daca $\text{inaltime}[5] > \text{maxim}$:

 - $1.9 > 1.7$ adevarat \Rightarrow acum $\text{maxim} = 1.9$ si $\text{poz_maxim} = 5$

Se verifica daca $\text{minim}(1.45) = \text{maxim}(1.9)$: fals, atunci se afiseaza Ioana si Dan

C++ daca doriti sa il verificati:

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    int n, poz_minim, poz_maxim;
    float minim, maxim;
    char nume[101][30];
    float inaltime[101];
    cin>>n;
    if(n==1){
        cout<<"Exista doar un elev.";
    }
    else{
        cin>>nume[1]>>inaltime[1];
        minim = inaltime[1];
        maxim = inaltime[1];
        poz_minim = 1;
        poz_maxim = 1;
        for(int i=2;i<=n;i++){
            cin>>nume[i]>>inaltime[i];
            if(inaltime[i] < minim){
                minim = inaltime[i];
                poz_minim = i;
            }
            if(inaltime[i] > maxim){
                maxim = inaltime[i];
                poz_maxim = i;
            }
        }
        if(maxim==minim)
            cout<<"Elevii au aceeasi inaltime";
        else
            cout<<nume[poz_minim]<<"\n"<<nume[poz_maxim];
    }
    return 0;
}
```

Testarea:

Se face cu cazurile speciale, bineînțeles. Asta înseamnă că dăm situațiile în care avem 1 elev, 2 elevi cu aceeași înălțime, 5 elevi cu aceeași înălțime și o situație obișnuită unde au de căutat în listă. Algoritmul poate fi oricare, chiar și cei mai „savanți”, atâta timp cât îndeplinește cerința.

Implementarea în limbaj de programare:

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    int n, poz_minim, poz_maxim;
    float minim, maxim;
    char nume[101][30];
    float inaltime[101];
    cin>>n;
    if(n==1){
        cout<<"Exista doar un elev.";
    }
    else{
        cin>>nume[1]>>inaltime[1];
        minim = inaltime[1];
        maxim = inaltime[1];
        poz_minim = 1;
        poz_maxim = 1;
        for(int i=2;i<=n;i++){
            cin>>nume[i]>>inaltime[i];
            if(inaltime[i] < minim){
                minim = inaltime[i];
                poz_minim = i;
            }
            if(inaltime[i] > maxim){
                maxim = inaltime[i];
                poz_maxim = i;
            }
        }
        if(maxim==minim)
            cout<<"Elevii au aceeași inaltime";
        else
            cout<<nume[poz_minim]<<"\n"<<nume[poz_maxim];
    }
    return 0;
}
```

Proiectarea algoritmului:

```
citește n
daca n==1 atunci
    scrie "Exista doar un elev"
altfel
    citește nume[1], inaltime[1]
    minim ← inaltime[1]
    maxim ← inaltime[1]
    poz_minim ← 1
    poz_maxim ← 1
    pentru i=2, i<=n executa:
        citește nume[i], inaltime[i]
        daca inaltime[i] < minim atunci
            minim ← inaltime[i]
            poz_minim ← i
        daca inaltime[i] > maxim atunci
            maxim ← inaltime[i]
            poz_maxim ← i
    daca maxim == minim atunci:
        scrie "Elevii au aceeași inaltime"
    altfel
        scrie nume[poz_minim], linie, nume[poz_maxim]
```