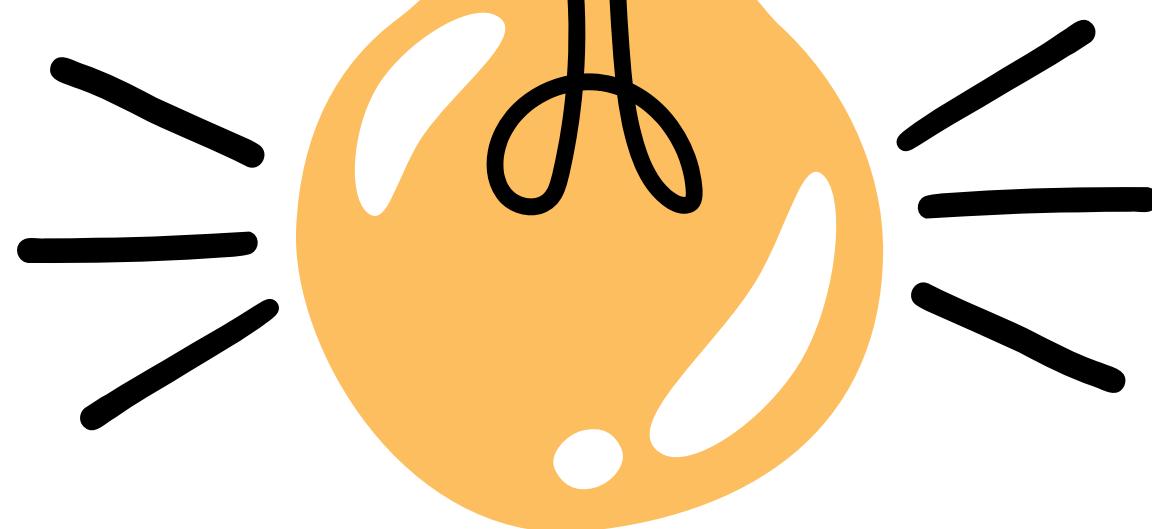
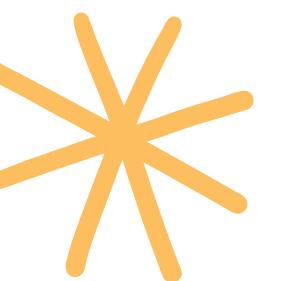
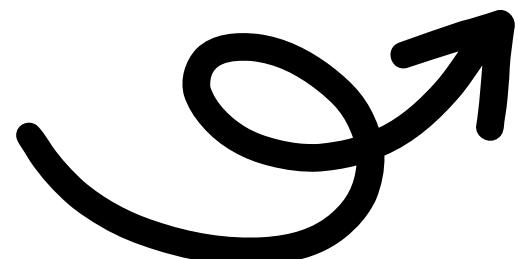




METODA

GREEDY



ee

PROBLEMA



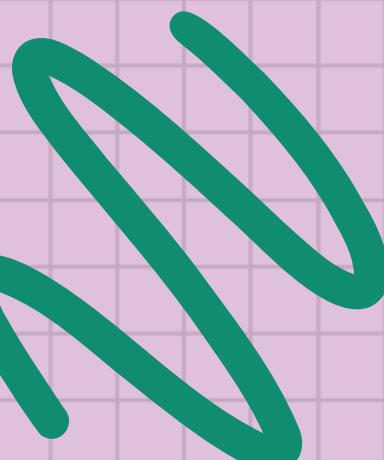
Se întâmplă de multe ori să vrem să cumpărăm multe cu cat de putine resurse.



Sau să vrem să facem de toate într-un timp scurt.

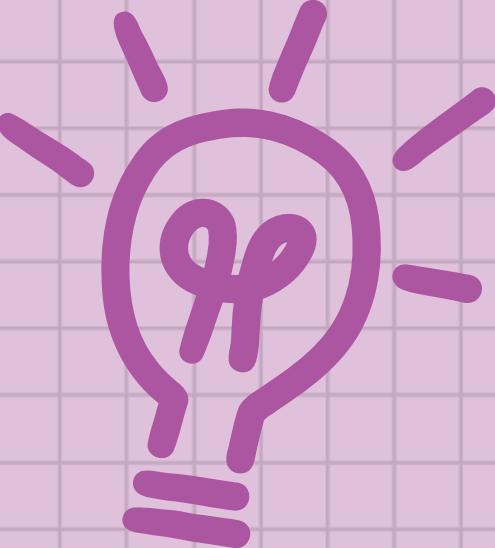


Sau poate chiar să ducem cu noi în ghiozdan cat de multe în excursie, dar spatele nostru duce o cantitate limitată de obiecte.





SOLUTIA: GREEDY



GREEDY IN VIATA COTIDIANA

CARE E NR. MAXIM DE OBIECTE PE
CARE LE PUTEM PUNE IN
RUCSAC?



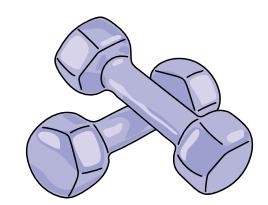
7 KG

1



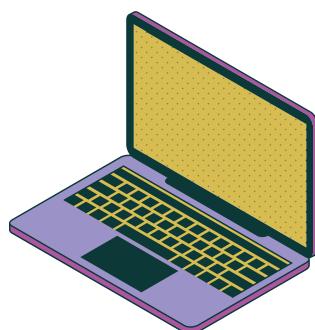
CATE: 10
MASA/BUC: 0,1 KG

2



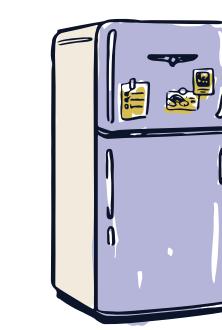
CATE: 2
MASA/BUC: 2 KG

3



CATE: 1
MASA/BUC: 3 KG

4



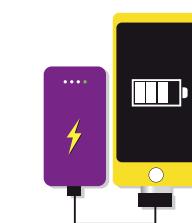
CATE: 1
MASA/BUC: 100 KG

5



CATE: 8
MASA/BUC: 0,6 KG

6



CATE: 1
MASA/BUC: 1 KG

7



CATE: 1
MASA/BUC: 20 KG

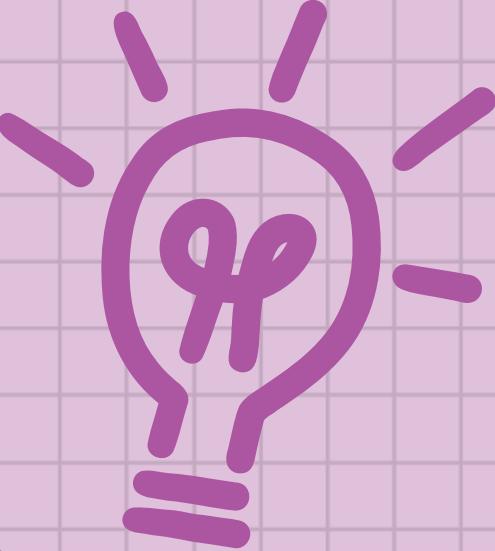
8



CATE: 1
MASA/BUC: 0,2 KG



SOLUTIA



ORDONARE

CARE E NR. MAXIM DE OBIECTE PE
CARE LE PUTEM PUNE IN
RUCSAC?



7 KG

1



CATE: 10
MASA/BUC: 0,1 KG

2



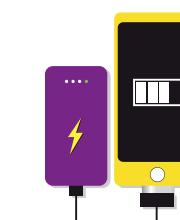
CATE: 1
MASA/BUC: 0,2 KG

3



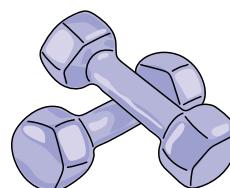
CATE: 8
MASA/BUC: 0,6 KG

4



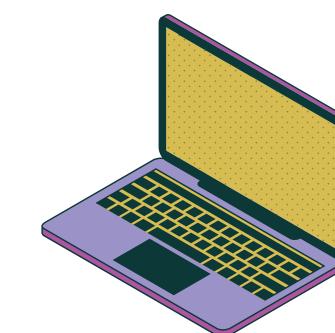
CATE: 1
MASA/BUC: 1 KG

5



CATE: 2
MASA/BUC: 2 KG

6



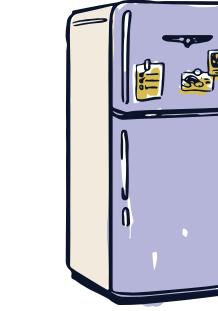
CATE: 1
MASA/BUC: 3 KG

7



CATE: 1
MASA/BUC: 20 KG

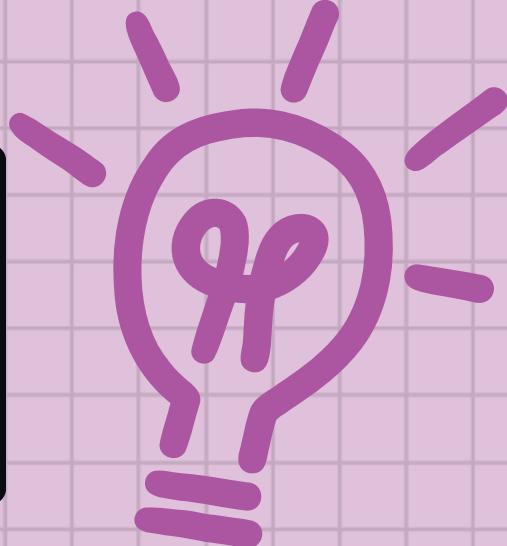
8



CATE: 1
MASA/BUC: 100 KG



SOLUTIA



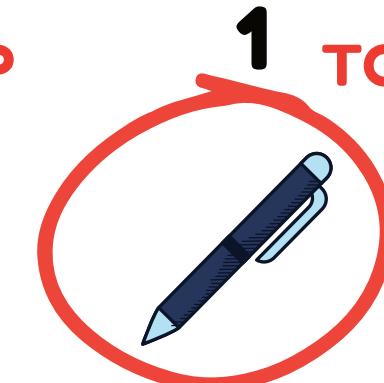
SELECTARE

ÎNTREBARE:

CE FACEM DACA NU ÎNCAP
TOATE PIXURILE?



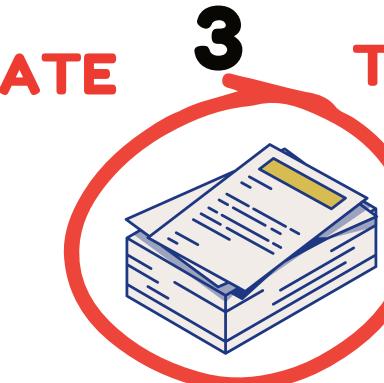
7 KG



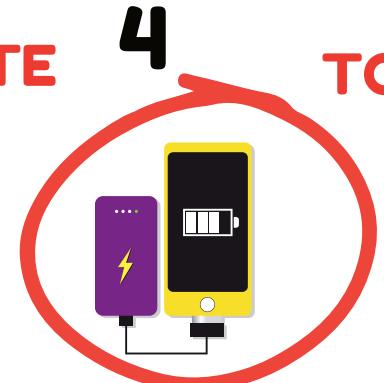
CATE: 10
MASA/BUC: 0,1 KG



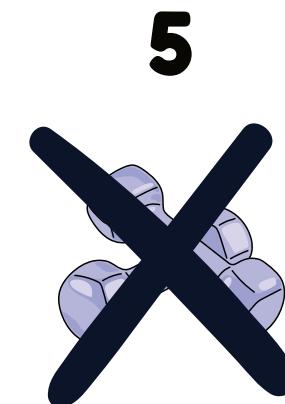
CATE: 1
MASA/BUC: 0,2 KG



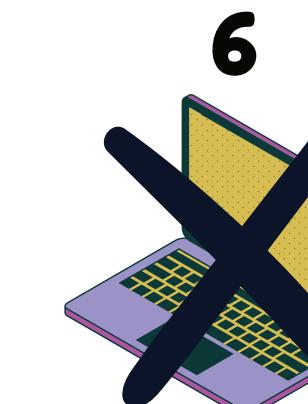
CATE: 8
MASA/BUC: 0,6 KG



CATE: 1
MASA/BUC: 1 KG



CATE: 2
MASA/BUC: 2 KG



CATE: 1
MASA/BUC: 3 KG



CATE: 1
MASA/BUC: 20 KG

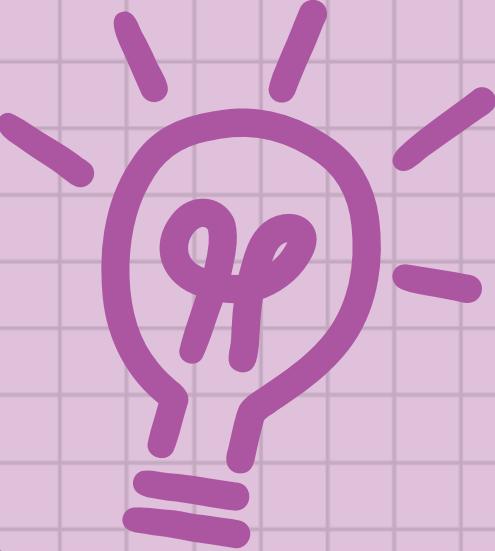
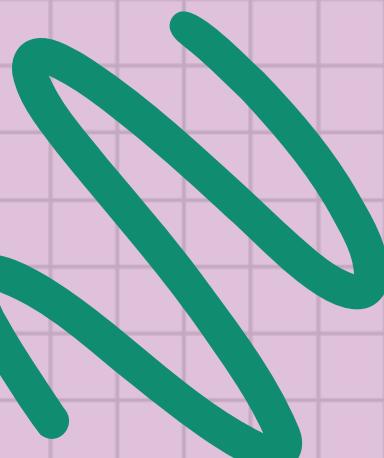


CATE: 1
MASA/BUC: 100 KG



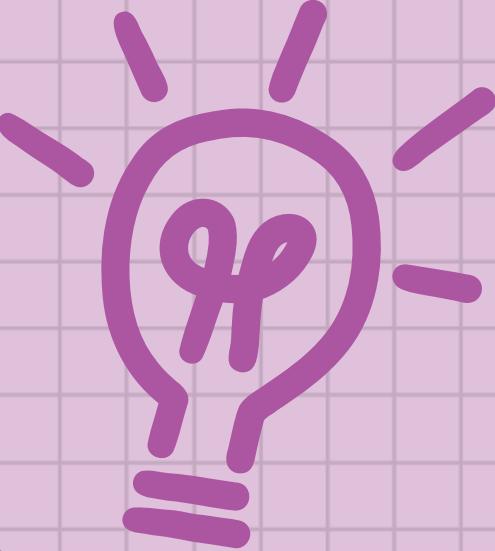
SA NE AMINTIM

CE TIPURI DE DATE CUNOASTEM?



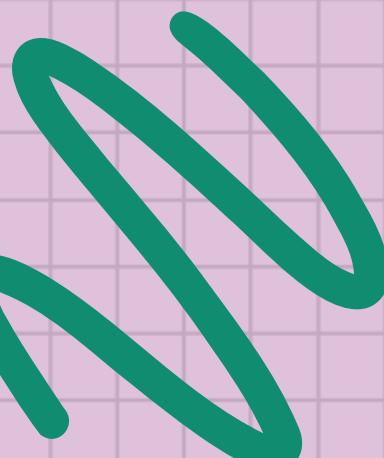


SA NE AMINTIM



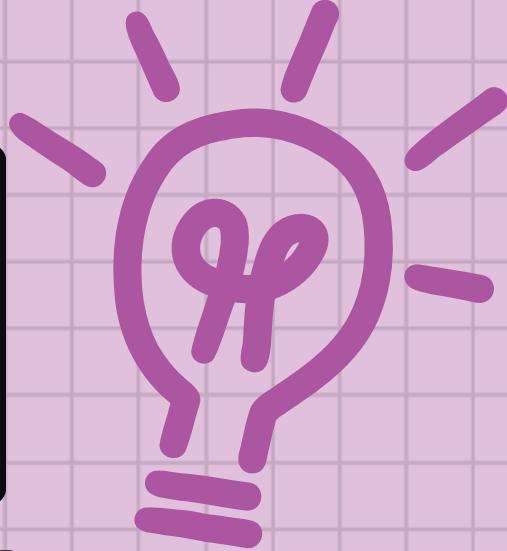
CE TIPURI DE DATE CUNOASTEM?

- Simple: int, float, bool, long
- Compuse: struct





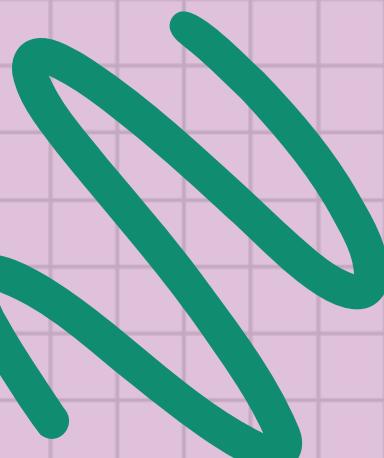
SA NE AMINTIM



CE TIPURI DE DATE CUNOASTEM?

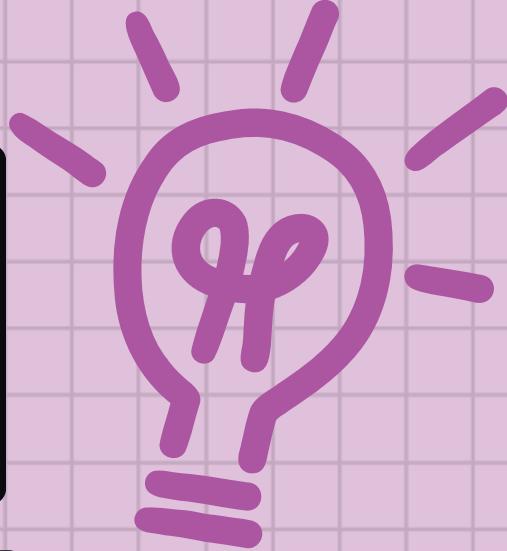
- Simple: int, float, bool, long
- Compuse: struct

CE TIPURI DE SORTARI CUNOASTEM?





SA NE AMINTIM

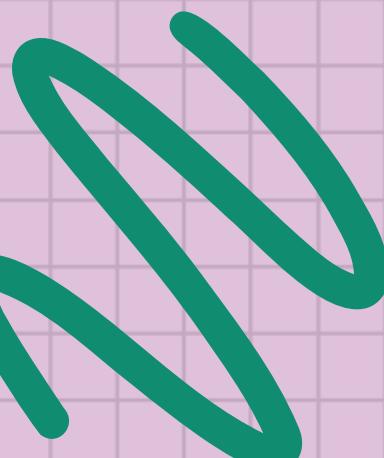


CE TIPURI DE DATE CUNOASTEM?

- Simple: int, float, bool, long
- Compuse: struct

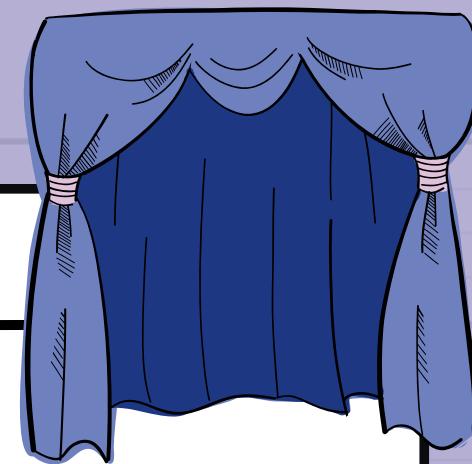
CE TIPURI DE SORTARI CUNOASTEM?

- Selection sort, Bubble sort, Insertion sort, Merge sort, Quick sort





Spectacole



Managerul artistic al unui festival trebuie să selecteze o mulțime cât mai amplă de spectacole ce pot fi jucate în singură sală pe care o are la dispoziție. Știind că i s-au propus $n \leq 100$ spectacole și pentru fiecare spectacol i-a fost anunțat intervalul în care se poate desfășura [si, fi) (si reprezintă ora și minutul de început, iar fi ora și minutul de final al spectacolului i)

Scriți un program care să permită spectatorilor vizionarea unui număr cât mai mare de spectacole.

De exemplu, dacă vom citi $n=5$ și următorii timpi:

12 30 16 30

15 0 18 0

10 0 18 30

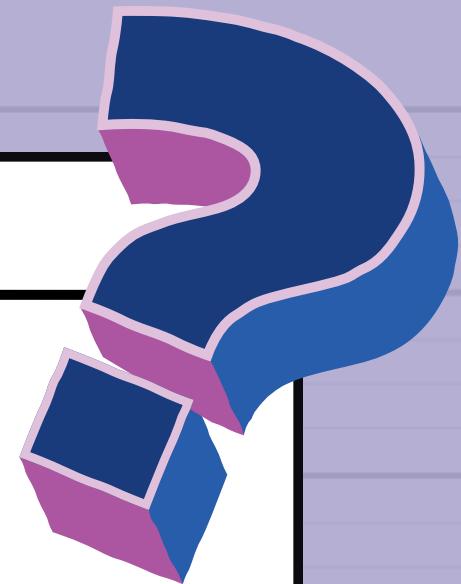
18 0 20 45

12 15 13 0

Spectacolele selectate sunt: 5 2 4.



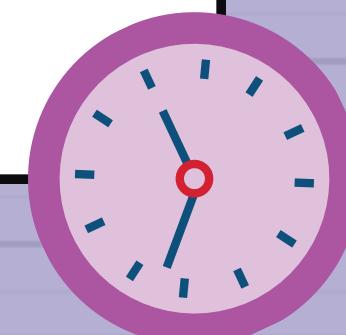
Întrebări



1) Ce și cum citim?

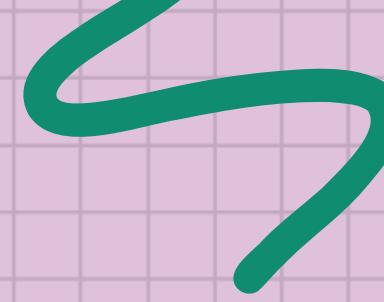
2) Cum ordonăm elementele și care?

3) Cum selectăm elementele?





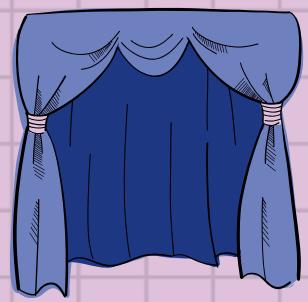
PASII



Citire



- Formăm o structură „timp”
- Formăm o structură „spectacol”
- Citim din fișier numărul de spectacole
- Citim din fișier spectacolele



Ordonare



- Ordonăm în funcție de ora de sfârșit
- Dacă orele de sfârșit sunt egale, ordonăm în funcție de minutul de sfârșit



Greedy-selectia



- Luăm câte elemente avem în ordinea în care am aranjat elementele în vector
- Ne asigurăm că ora de sfârșit a spectacolului nu se suprapune cu spectacolul la care suntem în vector.
- Facem afișarea în timpul parcurgerii.



REZOLVARE



1) CITIRE

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("spectacole.in");
struct timp
{ int ora,minut; };
struct spectacol {
    timp inc; // ora de inceput
    timp sf; // ora de sfarsit
    int nr; // nr de spectacole
} a[100]; // memorez spectacolele intr-un vector de
structuri
void citire(spectacol a[], int &n) {
    f>>n;
    for(int i=1; i<=n; i++){
        f>>a[i].inc.ora>>a[i].inc.minut;
        f>>a[i].sf.ora>>a[i].sf.minut;
        a[i].nr=i;
    }
}
```



2) ORDONARE

```
void ordonare(spectacol a[], int n)
{
    spectacol aux;
    for(int i=1; i<n; i++)
        for(int j=i; j<=n; j++)
            {if(a[i].sf.ora>a[j].sf.ora)//primul criteriu
            {
                aux=a[i];
                a[i]=a[j];
                a[j]=aux;
            }
            else if(a[i].sf.ora==a[j].sf.ora) //al doilea criteriu
                if(a[i].sf.minut>a[j].sf.minut)
                {
                    aux=a[i];
                    a[i]=a[j];
                    a[j]=aux;
                }
            }
}
```



REZOLVARE



3) GREEDY

```
void greedy(spectacol a[], int n){  
    cout<<"Lista de spectacole selectate este:"<<endl;  
    int nr=1; //cate spectacole am ales  
    afis(a[1]);  
    int ora_sf=a[1].sf.ora;  
    int min_sf=a[1].sf.minut;  
  
    for(int i=2; i<=n; i++)  
        if(a[i].inc.ora>=ora_sf && a[i].inc.minut>=min_sf ){  
            nr++;  
            afis(a[i]);  
            ora_sf = a[i].sf.ora; //pastrez ora de sfarsit a ultimului  
spectacol selectat  
            min_sf = a[i].sf.minut;  
        }  
  
    cout<<nr<<" selectat ca numar maxim de spectacole."  
<<endl;  
}
```



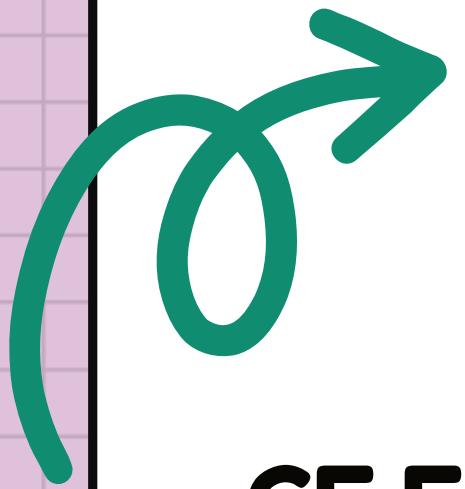
AFISARI

```
void afis(spectacol s){  
    cout<<"Spectacolul: "<<s.nr<<" Ora:"<<s.inc.ora<<":"<<s.inc.minut;  
    cout<<" - "<<s.sf.ora<<":"<<s.sf.minut;  
    cout<<endl;}  
  
void afisare(spectacol a[], int n){  
    for(int i=1; i<=n; i++)  
        afis(a[i]);  
    cout<<endl;}
```

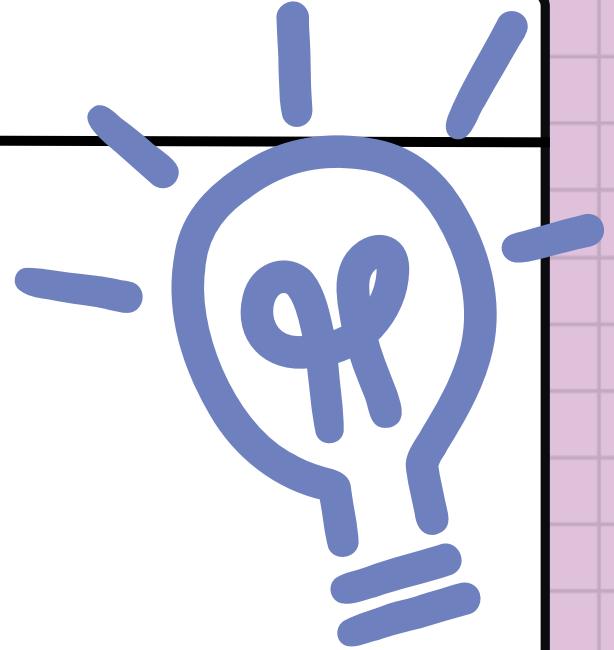


4) MAIN

```
int main(){  
    int n; citire(a,n);  
    cout<<"Lista de spectacole initiala este:"<<endl;  
    afisare(a,n);  
    ordonare(a,n);  
    cout<<"Lista de spectacole ordonate este:"<<endl;  
    afisare(a,n);  
    greedy(a,n);  
    return 0; }
```



CONCLUZII



CE ESTE GREEDY?

Un algoritm de a face alegeri mai potrivite in functie de nevoile noastre cu scopul de a economisi resurse si a eficientiza procesul de alegere.

CAND FOLOSIM?

Atunci cand avem mai multe elemente care trebuie sa imparta un aspect precum timpul sau spatiul.



CUM REZOLVAM?

Citim problema, citim datele, ordonam vectorul in functie de nevoile problemei si daca e cazul, calculam 'eficienta' unui produs, apoi selectam primele produse pana nevoia e indeplinita si nu se mai pot adauga elemente.



TEMA

PBINFO:

MOSCRACIUN2 #4010

BAL1 #3624

ARME1 #4335

