***Dura Alexandru-Bogdan***

***Grupa 163***

***Problema 1***

**Cerinta problemei**

**1.** Se dau N spectacole, pentru fiecare cunoscându-se identificatorul, ora de început și ora de sfârșit a spectacolului (ore care definesc intervalul de timp în care sala este rezervată, asigurându-se și intrarea/ieșirea spectatorilor) Să se determine un număr maxim de spectacole care pot fi programate în aceeași sală și în aceeași zi.

**Date de intrare:** fișierul date.in conține:

- pe prima linie un număr natural N, cu semnificația din enunț;

- pe următoarele N linii triplete de numere naturale de forma id start stop, unde id – identificatorul unui spectacol, start – ora de la care se poate intra în sală pentru acest spectacol, stop – ora la care s-a eliberat sala după acest spectacol.

**Date de ieșire:** fișierul date.out conține identificatorii spectacolelor selectate, în ordinea programării lor, câte unul pe rând.

|  |  |
| --- | --- |
| date.in | Date.out |
| 5  1 9 11  2 8 10  3 12 14  4 7 13  5 10 12   |  | | --- | |  | | 2  5  3 |

**Explicarea Algoritmului:**

- Am creat structura “Spectacol” care poate memora datele unui spectacol( ID – numarul spectacolului, start - ora la care incepe spectacolul si stop – ora la care se termina spectacolul)

-Am creat functia “compara” pe care o vom folosi in cadrul apelarii functiei qsort pentru a sorta in ordine crescatoare, spectacolele date in functie de ora la care se termina fiecare dintre acestea.

-Vom sorta spectacolele in functie de ora la care se termina deoarece astfel se poate ajunge la una dintre solutiile optime.

-Am citit din fisier datele pentru spectacolele date.

-Am planificat primul spectacol deoarece acesta este cel care se termina cel mai devreme.

-Apoi am verificat daca ora de incepere si spectacolele ramase este mai mare sau egala cu ora de finalizare a ultimului spectacol programatm iar daca indeplineste aceasta conditie, l-am adaugat la spectacolele programate.

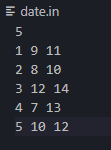
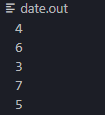
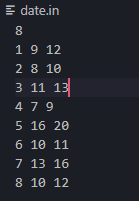
-La final am scris in fisierul “date.out” numerele spectacolelor programate in ordinea in care au fost programate.

Complexiatea Algoritmului:

Complexitatea algoritmului este O(n log n) datorita functiei qsort pe care am folosit-o pentru sortarea spectacolelor in ordine crescatoare in functie de ora la care se termina.

Cateva exemple :

Exemplul 1 Exemplul 2

Exemplul 3 Exemplul 4

