Cerinta:

Facultatea de Automatica si Calculatoare prin LSAC organizează in fiecare semestru un LAN Party la care participă toți studenții care vor sa își testeze abilitățile la un anumit joc. Se dorește realizarea unei soluții software(program executabil) ce poate automatiza acest proces.

Cerinte:

1. Se va crea o listă cu informațiile din fișierul d.in(lista se va forma prin adaugarea elementelor la inceput - O(1)).
2. Se vor elimina din lista anterior creată, echipele cu cel mai mic punctaj de echipă (informațiile despre modul de calcul al acestui punctaj se găsesc în cadrul secțiunii Detalii Tehnice) astfel incat in lista sa ramana n echipe (unde n maxim si n este putere a lui 2).În cazul în care mai multe echipe au acelasi scor se elimina prima echipă gasita in lista. Dupa fiecare eliminare se va relua parcurgerea listei de la cap.
3. Se vor implementa structurile de date necesare pentru a se efectua meciurile:

* Se va crea o coadă în care se vor pune meciurile dintre echipe.
* Echipele vor forma meciuri în ordinea în care se află în lista de la punctul anterior, primele două echipe vor forma primul meci (și așa mai departe).
* Se vor crea două stive - una pentru învinși și una pentru câștigători - ce vor conține echipele ce au pierdut, respectiv, au câștigat meciul anterior.
* Se vor introduce în stivă echipele în funcție de rezultatul obținut și se va adauga un punct la punctajul de echipă al echipelor ce au câștigat.
* In cazul unui meci in care ambele echipe au acelasi punctaj, se va alege prima echipa adaugata in meciul disputat.
* Se vor șterge echipele ce se găsesc în stiva de învinși.
* Din stiva de câștigători se vor extrage echipele și se va umple din nou coada de meciuri formându-se meciuri cu echipele consecutive (prima echipă scoasă din stivă va juca cu a doua echipa scoasă din stivă primul meci).
* Se vor repeta pașii enumerați pana la aflarea echipelor de pe primele 8 locuri, care vor fi salvate intr-o alta lista, dupa aflarea acestora se vor continua meciurile pana la aflarea echipei castigatoare.

Solutie:

Am ales sa implementez solutia problemei folosind pentru fiecare structura de date ce trebuia adaugata cate o clasa. Fiecare clasa are un header si un .cpp care contin metodele necesare pentru rezolvarea exercitiului.

Consider ca tema este utila in a verifica atat cunostintele de Programare Orientata pe Obiecte cat si cele de Structuri de Date si Algoritmi.

Din punctul meu de vedere, pentru subpunctele pe care am reusit sa le fac, implementarea este una eficienta.

Implementare:

In clasa player:

Pe langa constructori si functia de afisare am facut si get-eri pentru primul si al doilea nume.

In clasa team:

Pe langa constructori, redefinirea operatorului egal, si destructor(necesare avand un atribut de tip pointer), functia de afisare si de citire in/din fisier;

1.o functie care imi calcula punctajul per echipa

2.o functie prin care puteam seta punctajul

3.o functie care imi returna toti jucatorii

4. o functie care return numele echipei respective

In clasa lista:

Pe langa constructor si destructor,functia de adaugare si stergere nod si afisarea

1. o functie pentru afisarea numelor echipelor din lista

2. o functie de gasire a minimului care returneaza echipa cu punctaj minim din lista

3. o functie care returna capul de lista

In clasa game:

Constructorii destructorul functie de afisare, redefinirea operatorului egal, get-eri pentru echipe

Am implementat, de asemenea:

1.functie pentru a extrage castigatorul meciului in functie de scorul obtinut de fiecare echipa

2.functie de extragere a pierzatorului

In coada:

Constructor destructor functia de adaugare si stergere meci, afisarea cozii si un get-er pentru capul cozii.

Pentru coada am impementat o alta clasa header numita nodeQ.

In Stack:

Constructorii, destructorul, afisarea

1.o functie push pentru adaugarea de echipe la inceputul structurii de date

2.o functie pop pentru stergearea ultimului element

3.o functie pentru afisarea link-ului

In main se regaseste solutia problemelor propuse in cadrul cerintei de la inceputul paginii.