

# Tema 1 - Structuri de date (seria CB)

## Poker Tournament

Responsabili temă:	Busioc Costin, Mitran Andreea, Neacșu Bogdan Ștefan
Data publicării:	<b>11.03.2018</b>
Termenul de predare:	<b>2.04.2018 (ora 23:55)</b>  Se accepta teme trimise cu penalizare de 10 puncte / zi (din maxim 100 puncte) pana la data de <b>5.04.2018 ora 23:55</b> .

## 1. Introducere

Turneele de poker se pot organiza cu un număr aproape nelimitat de jucători. Cu toate acestea, organizatorii au sarcina de a stabili un algoritm prin care aceștia să fie împărțiți la mese. Mai mult, pe parcursul turneului, jucătorii pot pierde jocul și trebuie să-și elibereze locul, care poate fi folosit de un alt jucător. Iar pentru a păstra jocul de cărți interesant, trebuie să existe cât mai mulți jucători la o masă. De aceea organizatorii pot decide închiderea unei mese și redistribuirea jucătorilor acesteia la celelalte mese. Pentru a face față acestei munci titanice, organizatorii vă cer ajutorul pentru realizarea unui turneu reușit.

## 2. Cerință

Tema presupune să realizați un program care să ajute la organizarea și așezarea jucătorilor la un turneu de poker.

## 3. Implementare

**Sala** de joc va fi reprezentată sub forma unei liste simplu înlănțuite **fără santinelă (nu este circulară)**, conținând elemente de tip TMasa. Structura TMasa este descrisă în scheletul de cod primit la începerea temei.

O **masa** de joc este reprezentată sub forma unei liste **circulare** simplu înlănțuite **cu santinelă**, a cărei structură a fost menționată și mai sus. Aceasta va avea în componență numele mesei (char \*), numărul curent de jucători (int), numărul maxim de jucători (int) și o listă de elemente de tip jucător, structura care de asemenea va fi definită în scheletul de cod.

Jucătorul va fi definit printr-un id de identificare (char \*) și un număr de mâini pe care le mai poate juca la momentul curent (int).

Implementarea trebuie să înceapă de la header-ul (info.h) primit în arhiva temei. **Nu este permisă** modificarea structurilor existente!

Listele utilizate în rezolvarea temei trebuie să fie **GENERICE**. Nerespectarea genericității și implementarea temei folosind liste particulare (pentru fiecare structura în parte) va duce la o depunctare totală de **30 DE PUNCTE**.

Programul va primi drept argumente în linia de comandă 3 fișiere cu următoarea semnificație:

- configuratie.txt - configurația inițială a meselor și informații despre jucători
- evenimente.txt - înălțuirea de evenimente ce avea loc asupra sălii de joc. Acestea vor fi detaliate mai jos
- out.txt - output-ul cerut pentru afișare în urma diferitelor evenimente.

Evenimentele ce pot avea loc asupra sălii/meselor/jucătorilor sunt:

**MENȚIUNE:** Căsuța etichetată cu valoarea "S" în desenele ce urmează, reprezintă santinela listei respective.

#### a) print

Evenimentul prin care se va face afișarea sălii la momentul curent. Mesele vor fi afișate în ordinea primită în fișierul inițial (configuratie.txt), iar jucătorii de la masă vor fi afișați în ordinea mâinii de joc curente (configurația la momentul actual). Jucătorul care se află "primul" la momentul curent este chiar cel din dreapta santinelei.

Formatul de afișare este următorul:

Nume\_masa\_1: nume\_jucător\_1 - număr\_mâini\_jucător\_1; nume\_jucător\_2 - număr\_mâini\_jucător\_2; .....; nume\_jucător\_n1 - număr\_mâini\_jucător\_n1.

Nume\_masa\_2: nume\_jucător\_1 - număr\_mâini\_jucător\_1; nume\_jucător\_2 - număr\_mâini\_jucător\_2; .....; nume\_jucător\_n2 - număr\_mâini\_jucător\_n2.

.....

Nume\_masa\_N: nume\_jucător\_1 - număr\_mâini\_jucător\_1; nume\_jucător\_2 - număr\_mâini\_jucător\_2; .....; nume\_jucător\_nN - număr\_mâini\_jucător\_nN.

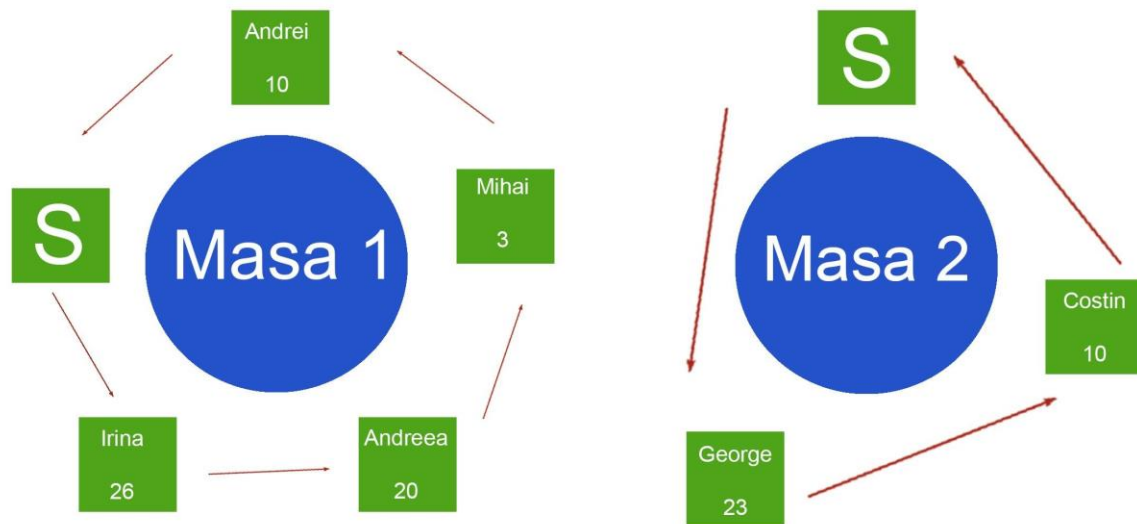
unde n1, n2, .., ni, .. nN - reprezintă numărul de jucători de la masa i

N - reprezintă numărul de mese.

Fiecare masă se va afișa pe câte o linie separată!

**ATENȚIE:** În cazul în care nu există nicio masă, va fi afișat mesajul "*Sala este inchisă!*"

**Exemplu:**



**Output:**

Masa1: Irina - 26; Andreea - 20; Mihai - 3; Andrei - 10.

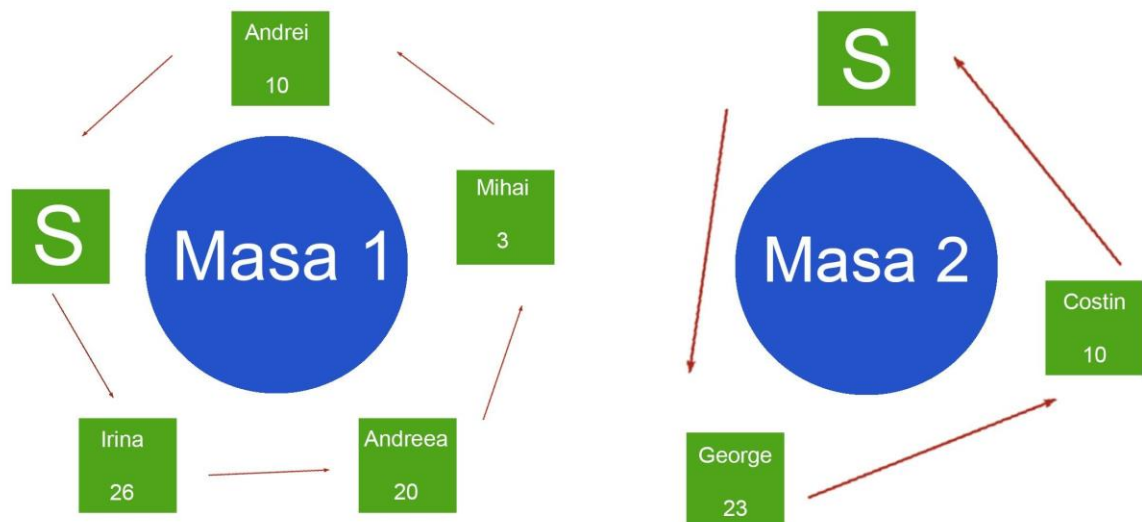
Masa2: George - 23; Costin - 10.

**b) noroc [nume\_masă] [nume\_jucător] [grad\_noroc]**

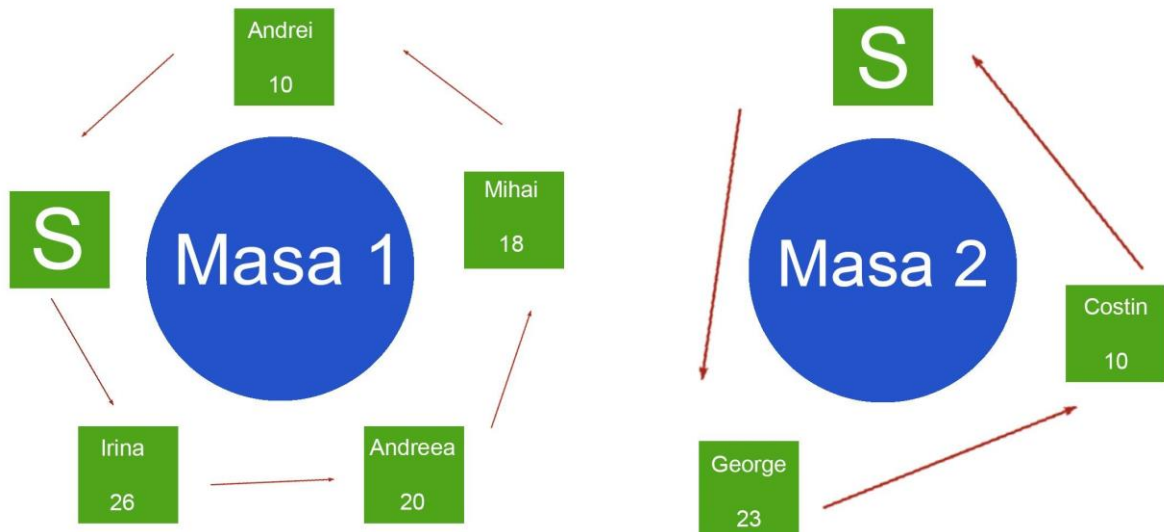
Evenimentul prin care un jucător primește mai multe șanse de a rămâne în joc, numărul de mâini pe care acesta le poate juca fiind incrementat. Parametrii reprezintă:

- noroc - numele operației
- nume\_masa - numele mesei la care jucătorul se află
- nume\_jucător - numele jucătorului "norocos"
- grad\_noroc - întreg mai mare ca 0

**Exemplu:**



Aplicare eveniment: **“noroc Masa1 Mihai 15”**

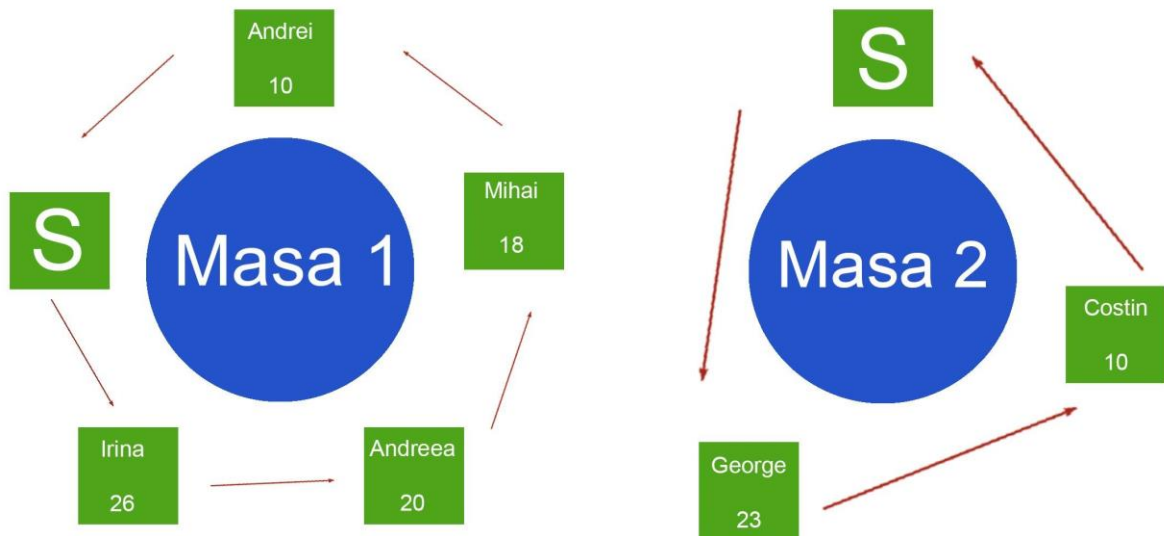


**c) ghinion [nume\_masă] [nume\_jucător] [grad\_ghinion]**

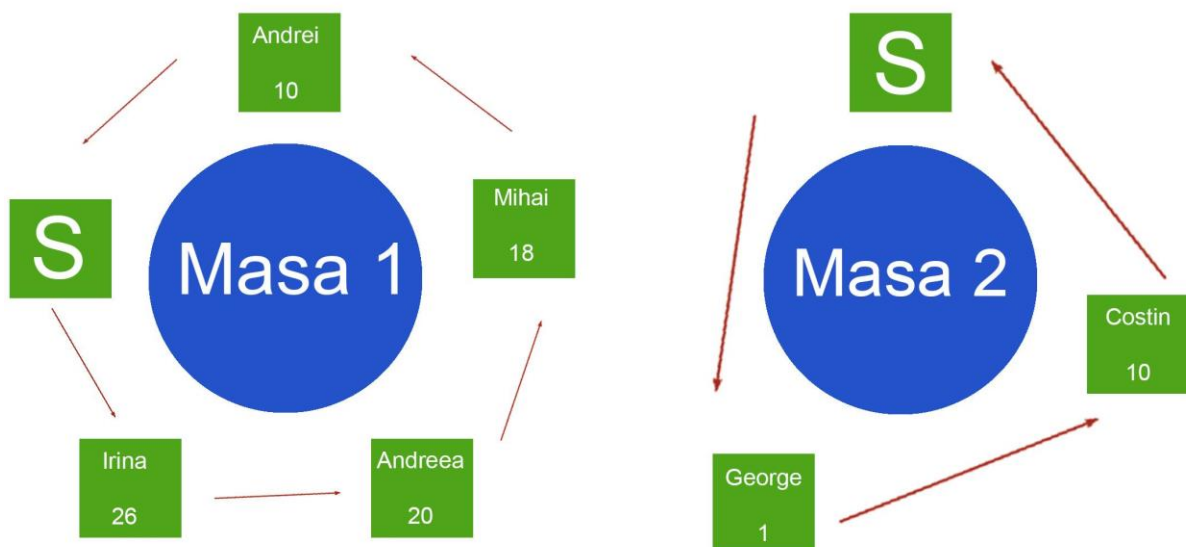
Este evenimentul opus precedentului (noroc), prin care unui jucător i se scade din numărul de mâini pe care le poate juca valoarea grad\_ghinion. Comportamentul evenimentului este similar celui precedent, se vor afișa aceleași mesaje de eroare în

cazul în care datele sunt invalide, iar altfel va avea loc o simplă modificare a datelor, fără a se scrie în fișierul de output.

**Exemplu:**

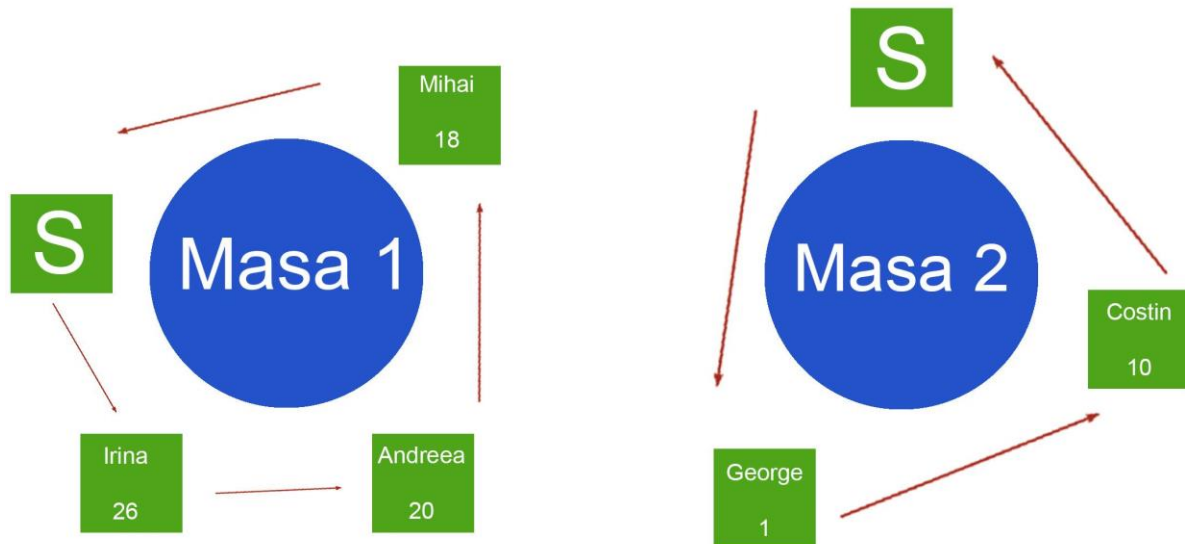


Aplicare eveniment: **“ghinion Masa2 George 22”**





Aplicare eveniment: “ghinion Masa1 Andrei 15”



De asemenea, trebuie verificat pentru ambele evenimente (noroc, ghinion) faptul că datele sunt valide. Asta înseamnă că:

- În cazul în care numele mesei nu este valid ([nume\_masă] nu apare în lista de mese) se va afișa pe o linie separata în fișierul de output mesajul: “Masa [nume\_masa] nu exista!”
- În cazul în care numele jucătorului nu este valid ([nume\_jucător] nu apare în lista mesei [nume\_masa]) se va afișa pe o linie separata în fișierul de output mesajul: “Jucatorul [nume\_jucător] nu exista la masa [nume\_masa]!”
- În cazul în care datele sunt valide, atât [nume\_masa], cat și [nume\_jucător] exista, nu se va afișa nimic în fișierul de output, însa datele vor fi modificate corespunzător.

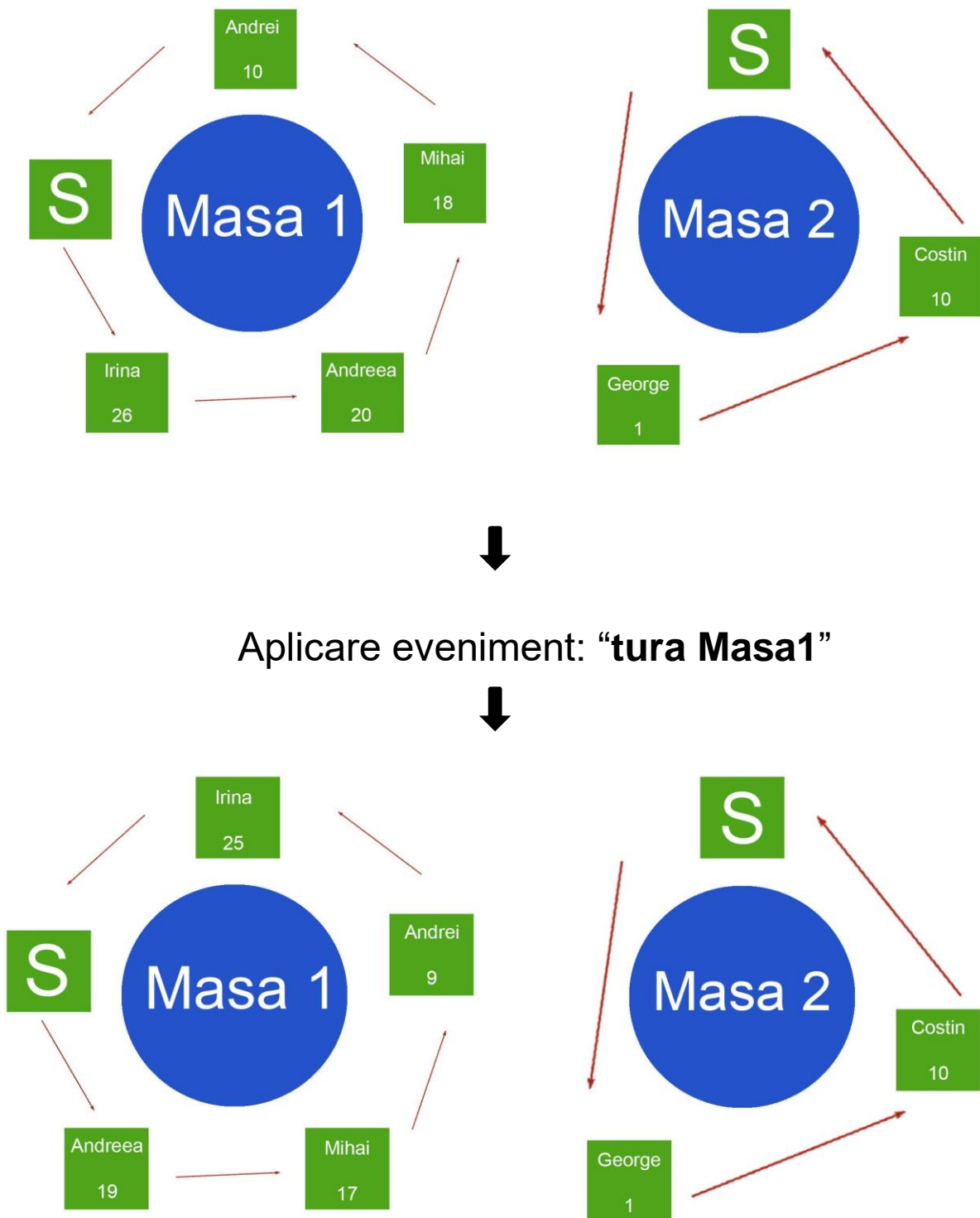
#### d) tura [nume\_masă]

Evenimentul prin care se informează că a avut loc o mână jucată la o anumită masa. Acest lucru implică decrementarea cu 1 a valorii din câmpului număr de mâini pentru fiecare jucător de la masa respectivă și reordonarea în listă conform regulii: al

doilea jucător (al doilea element din lista) devine primul; al treilea devine al doilea;... primul jucător devine ultimul. Rotația are loc în jurul santinelei.

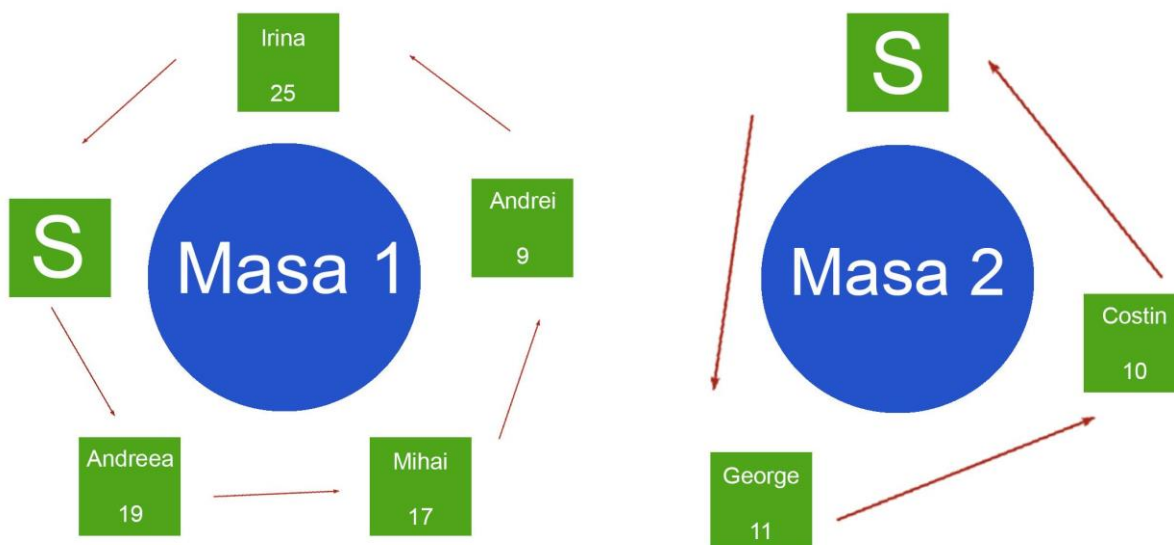
În urma evenimentului, nu va fi afișat nimic în fișierul de ieșire. Cu toate acestea, în cazul în care [nume\_masă] este invalid, se va scrie mesajul: “Masa [nume\_masa] nu exista!”

**Exemplu:**

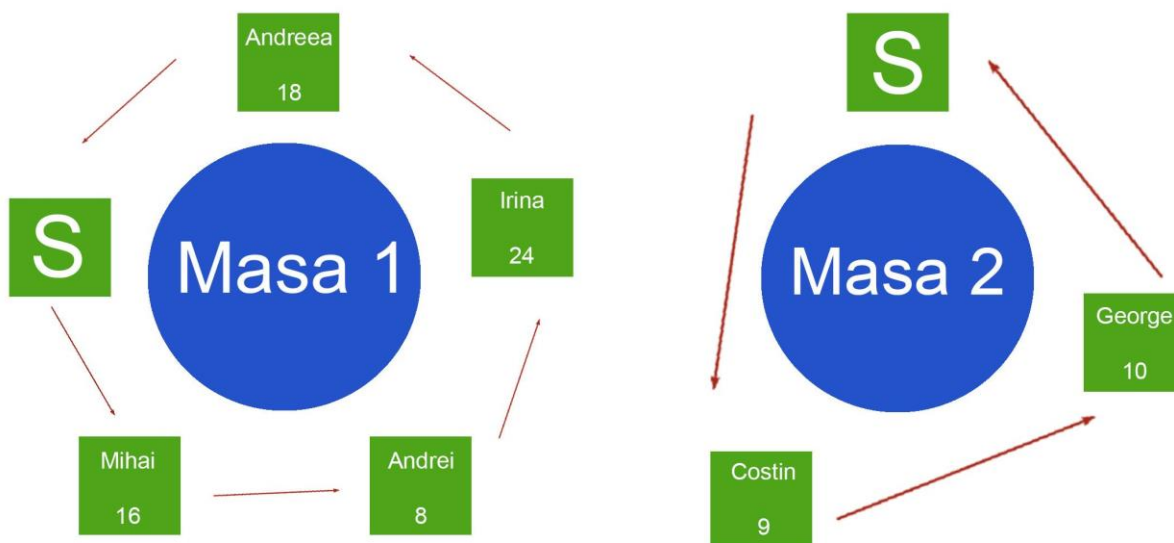


**e) tura\_completa**

Evenimentul are un comportament similar celui descris mai sus, însă modificarea va fi aplicată pentru toate mesele la acest moment.



Aplicare eveniment: **“tura\_completa”**





**ATENȚIE:** În cazul în care în urma unui eveniment, un jucător ajunge să aibă valoarea numărului de mâini pe care le poate juca mai mică sau egală cu 0, jucătorul va fi eliminat de la masa (înregistrarea sa va fi ștearsă din lista).

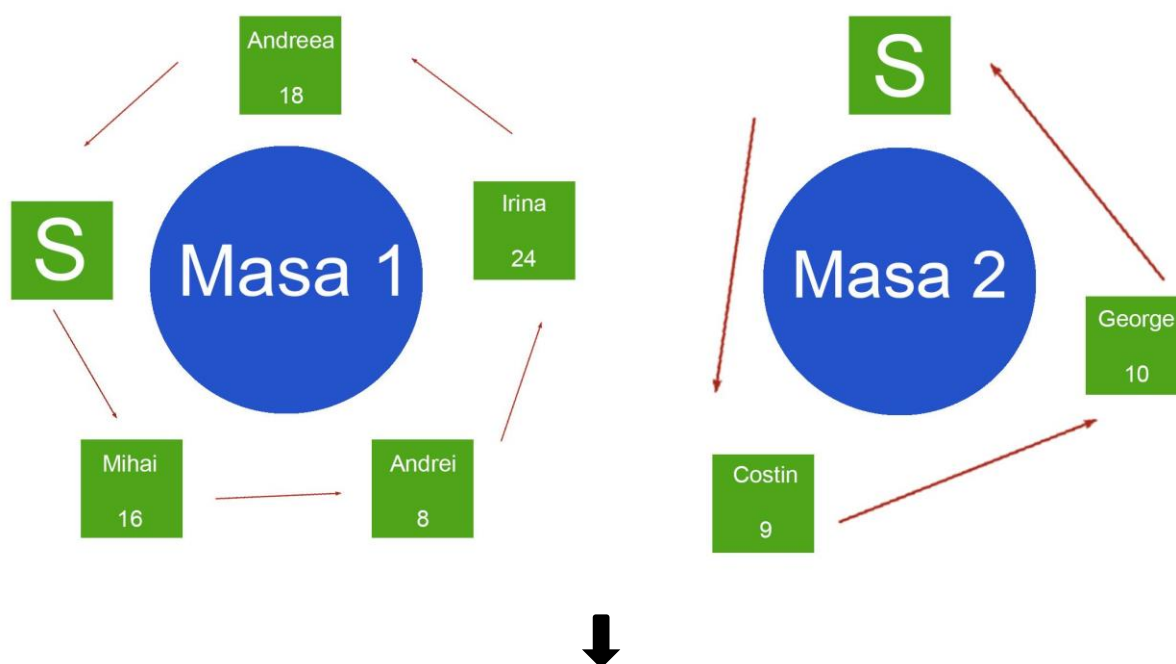
**ATENȚIE:** În cazul în care masa are un număr curent de jucători egal cu 0, ea va fi eliminată și ștearsă din memorie. (Implicit, nu va mai fi afișată la operația print).

#### f) clasament [nume\_masă]

Prin apelarea acestui eveniment, se va afișa în fișierul de output un clasament al jucătorilor de la o anumită masă, descrescător după numărul de mâini rămase pentru a fi jucate de fiecare jucător. Se va afișa pe câte o linie numele jucătorului, urmat de numărul de mâini rămase (separate prin spațiu) după cum urmează, începând cu string-ul "Clasament nume\_masa:". **Mențiuni sortare:**

- Sortarea se va face utilizând o noua lista. Lista va fi construită sortată descrescător în funcție de clasamentul jucătorilor, prin inserarea ordonata, succesiva jucătorilor.
- În cazul în care doi jucători au același număr de mâini rămase pentru a fi jucate, compararea dintre cei doi se va face lexicografic, în funcție de numele acestora, folosind ordinea din tabela ASCII. (ex.: "Ion" < "Ionela", "Andrei" < "Bogdan", "Ion" > "IoN")
- În cazul în care afișarea clasamentului se realizează prin construirea unui vector și sortarea acestuia, soluția pentru testul respectiv va fi punctată la jumătate din valoarea sa.

#### Exemplu:



## Aplicare eveniment: “clasament Masa1”



out.txt

```
Clasament Masa1:  
Irina 24  
Andreea 18  
Mihai 16  
Andrei 8
```

Similar, în cazul în care [nume\_masă] este invalid, se va afișa același mesaj: “Masa [nume\_masa] nu exista!”

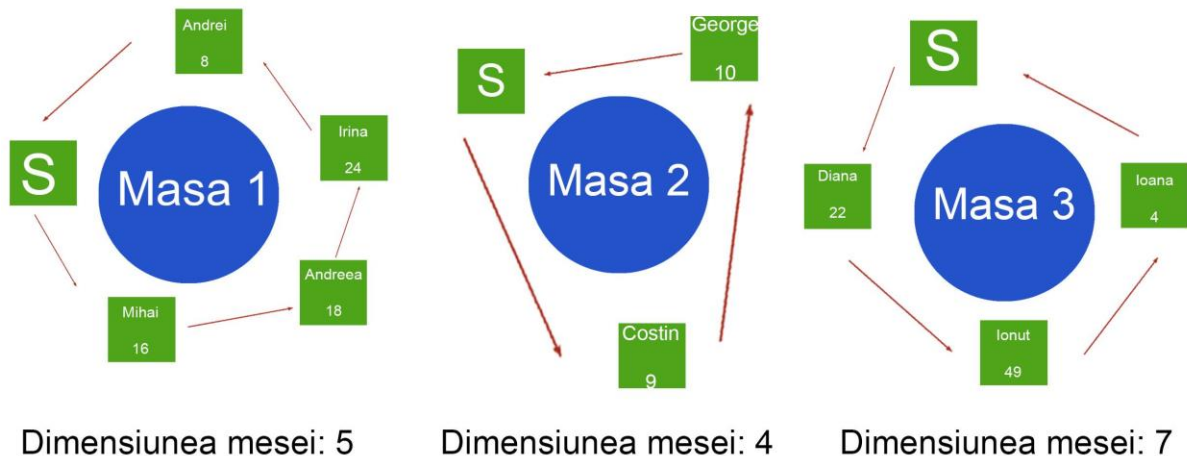
### g) inchide [nume\_masa]

Apelând evenimentul “îchide nume\_masa”, se dorește desființarea (ștergerea) mesei primita ca parametru. Dacă [nume\_masa] este invalid, se va afișa același mesaj: “Masa [nume\_masa] nu exista!”. Altfel, lista destinată acelei mese va fi ștearsă din memorie, iar jucătorii aflați la masa respectiva vor fi redistribuiți la mese după cum urmează:

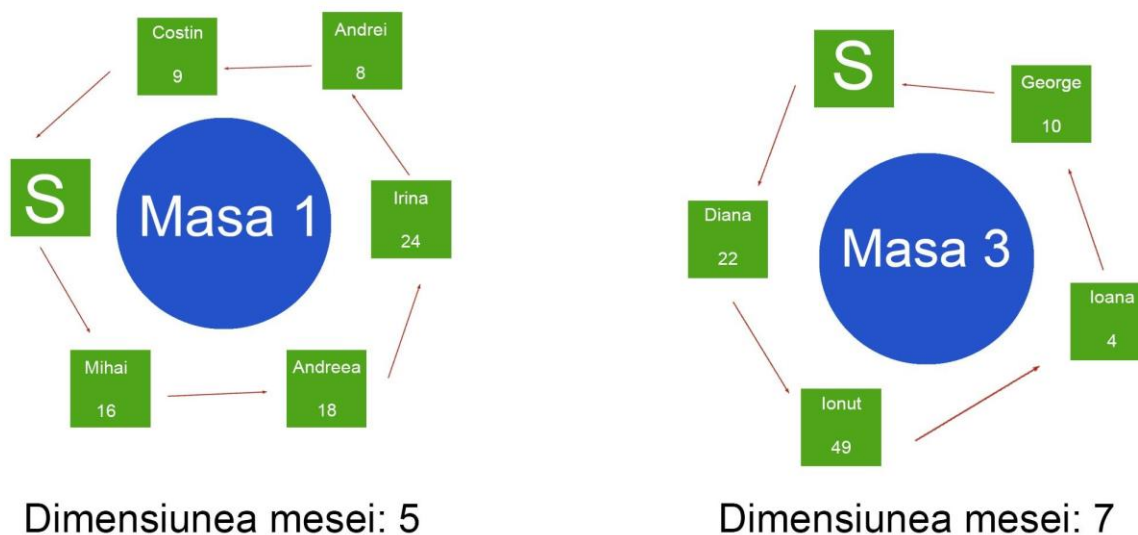
- Se vor parcurge mesele de la început spre sfârșit (în ordinea meselor primite în fișierul de configurație)
- Dacă masa curentă dispune de locuri libere, vor fi adăugați atâția jucători de la masa eliminată încât noua masă permite, în ordinea în care aceștia se aflau la masa închisă. Jucătorii vor fi adaugați la sfârșitul listei
- Procedul se repetă, continuând la următoarea masă, până când toți jucătorii de la masa eliminată vor fi așezați la o noua masă

**ATENȚIE:** În cazul în care nu există suficiente locuri în sală pentru jucătorii de la masa care se închide, se va scrie în fișier mesajul “Nu exista suficiente locuri în sala!”, structura sălii fiind nemodificată în acest caz.

**Exemplu:**



Aplicare eveniment: “**închide Masa2**”



**Explicație:** După cum se observă, Masa1 mai dispune de un singur loc disponibil. Costin este primul jucător de la masa ce va fi eliminată așa că el va obține acel loc. Cel de-al doilea jucător, George, va fi așezat la Masa3, unde după cum se observă sunt suficient de multe locuri ramase.

## 4. Rulare

Programul v fi rulat astfel:

**./tema1 configuratie.txt evenimente.txt out.txt**

unde:

- configuratie.txt - fișierul de configurare al sălii și meselor
- evenimente.txt - fișierul de evenimente produse asupra sălii
- out.txt - fișierul cu date de ieșire
- Fișierul evenimente.txt va conține fiecare comanda pe o linie noua
- Fișierul configuratie.txt va fi descris mai jos
- Tema se va implementa folosind liste simplu înlănțuite **generice**.

Exemplu **configuratie.txt**:

```
N
nume_masa_1 n1 dim1
nume_jucator_1 număr_mâini
nume_jucator_2 număr_mâini
....
nume_jucator_n1 număr_mâini
nume_masa_2 n2 dim2
nume_jucator_1 număr_mâini
....
nume_jucator_n2 număr_mâini
.....
nume_masa_N nN dimN
nume_jucator_1 număr_mâini
....
nume_jucator_nN număr_mâini
```

unde:

- N - numărul de mese din sala
- pe următoarea linie va fi numele primei mese, numărul de jucători de la masa (n1) și dimensiunea mesei (dim1)
- pe următoarele n1 mese se vor afla linii de forma: numele\_jucătorului și numărul său de mâini, separate prin spațiu
- procedeul se repeta pentru încă N-1 mese

## 5. Notare

- **85 puncte** obținute pe testele de pe **vmchecker**;
- **10 puncte: coding style**, codul trebuie sa fie comentat, consistent și ușor de citit (a se vedea [1]). De exemplu, tema nu trebuie sa conțină:
  - warninguri la compilare;
  - linii mai lungi de 80 de caractere;
  - tab-uri amestecate cu spatii;
  - denumire neadecvata a funcțiilor sau a variabilelor;
  - folosirea incorectă de pointeri, neverificarea codurilor de eroare;

- utilizarea unor metode ce consumă resurse în mod inutil (alocare de memorie);
- neeliberarea resurselor folosite (eliberare memoriei alocate, ștergerea fișierelor temporare, închiderea fișierelor);
- alte situații nespecificate aici, dar considerate inadecvate;
- **5 puncte: README** – va conține detalii despre implementarea temei, precum și punctajul obținut la teste (la rularea pe calculatorul propriu)
- **Bonus: 20 puncte** pentru soluțiile ce nu au memory leak-uri (bonusul se va considera numai în cazul în care a fost obținut punctajul aferent testului)
- **Temele care nu compilează, nu rulează sau obțin punctaj 0 la teste, indiferent de motive, vor primi punctaj 0**

## 6. Reguli de trimitere a temelor

- Temele vor fi încărcate pe **vmchecker** (în secțiunea Structuri de Date seria CB: SD-CB), dar și pe **cs.curs.pub.ro**, în secțiunea destinată assignment-ului “Tema1”
- Arhiva finală a temei rezolvate trebuie să conțină:
  - fișierele sursă
    - Fiecare fișier sursă creat sau modificat trebuie să înceapă cu un comentariu de forma:  
**/\* NUME Prenume - grupa \*/**
  - fișierul README în care va fi detaliat modul de implementare al rezolvării
  - fișierul Makefile cu două reguli (build și clean)
    - Fișierul trebuie obligatoriu denumit Makefile și trebuie să conțină cele 2 reguli menționate
    - Regula build va compila sursele și va crea executabilul numit tema1
    - Regula clean care va șterge executabilele create
  - Arhiva va conține numai fișierele menționate mai sus (nu se acceptă fișiere executabile sau obiect)
  - Dacă arhiva nu respectă aceste specificații, aceasta nu va fi acceptată la upload și implicit tema nu va fi luată în considerare
- A se consulta și Regulile generale de trimitere și punctare a temelor [2]

## 7. Referințe

[1] <https://ocw.cs.pub.ro/courses/programare/coding-style>

[2] <http://cs.curs.pub.ro/> - curs SD - seria CB - secțiunea Regulament SD - Reguli de realizare, verificare și trimitere a temelor