

RESTANȚĂ LA DISCIPLINA “PROGRAMARE PROCEDURALĂ”

- SERIILE 13 ȘI 14, Luni, 4 iunie 2018 -

Pe o tablă de joc de diferite forme (mai întâi vom considera table de forma unui pătrat $n \times n$) doi jucători pun succesiv bile marcate cu X (primul jucător) și O (al doilea jucător). Primul jucător (cu X) este primul la mutare. Bila curentă a unui jucător e pusă pe o coloană și cade pe linia cea mai de jos disponibilă. După ce tabla se umple de bile (n^2 bile – pe cazul tablei sub formă de pătrat) se calculează scorul unui jucător ca fiind lungimea maximă pe orizontală sau verticală a unei secvențe consecutive de X (pentru primul jucător) respectiv de O (pentru al doilea jucător).

Exemplu: considerăm $n = 5$ (tabla are 5 linii și 5 coloane, 25 de poziții pe care se pot pune bile) și specificăm mutările jucătorilor prin enumerarea coloanelor pe care se pun bilele. Dacă aceste coloane sunt: 1, 2, 3, 4, 4, 5, 1, 5, 2, 3, 2, 3, 3, 1, 2, 5, 1, 5, 5, 4, 4, 4, 3, 2, 1 atunci configurația tablei pe parcursul jocului arată astfel:

			X	
X	O	X	O	

după a 5-a mutare

X	X	O	X	O
X	O	X	O	O

după a 10-a mutare

X	O	X	O	X
X	X	X	X	O
O	X	O	O	O
X	X	O	X	O
X	O	X	O	O

după ultima mutare

Scorul primului jucător este 4 (lungimea secvenței orizontale de culoare roșie de X), scorul celui de-al doilea jucător este 4 (lungimea secvenței verticale de culoare albastră de O).

Fișierele text cu numele *tablaPatrata_n.txt* au pe prima linie valoarea lui n (pentru cazul unei table pătrate) și apoi pe fiecare dintre cele n^2 linii următoare mutările celor doi jucători (codificate prin coloana pe care se pune bila). Fișierul *tablaPatrata_5.txt* conține datele pentru exemplul de mai sus.

Cerințe:

- Scriveți o funcție cu numele **genereazaTablaPatrata** care creează un tablou bidimensional pătratic (tabla de joc) cu n linii și n coloane alocat dinamic. **(1 punct)**
- Scriveți o funcție cu numele **determinaConfiguratieTablaPatrata** care determină și afișează pe ecran configurația finală a tablei de joc pătrate pe baza unui fișier de intrare de forma **tablaPatrata_n.txt**. În scrierea funcției puteți să considerați returnarea unei structuri care să conțină valoarea lui n și pointerul către tabloul alocat dinamic. **(2.5 puncte)**
- Scriveți o funcție cu numele **calculeazaScoruriTablaPatrata** care calculează și returnează scorurile celor doi jucători pe baza configurației finale a tablei pătrate de joc. **(2 puncte)**
- Scriveți un program care prin apeluri ale funcțiilor de la punctele a-c afișează scorurile celor doi jucători pentru fișierul **tablaPatrata_5.txt** din arhivă. **(0.5 puncte)**

Considerăm acum cazul unei table de joc de formă triunghiulară ca mai jos:

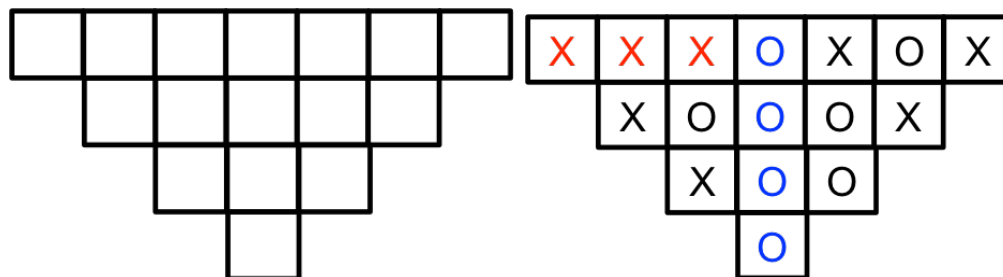


Tabla de joc are n coloane (n este impar). Coloanele i , cu i între 1 și $(n-1)/2$ au i poziții (linii) disponibile spre a fi plasate bile ca în figura de mai sus. Coloanele i , cu i între $(n-1)/2 + 1$ și n au $n-i+1$ poziții (linii) disponibile spre a fi plasate bile. Regulile jocului prezentate anterior se păstrează. După ce tabla se umple de bile $((n+1)^2/4$ bile – pe cazul tablei triunghiulare) se calculează scorul unui jucător ca fiind lungimea maximă pe orizontală sau verticală a unei secvențe consecutive de X (pentru primul jucător) respectiv de O (pentru al doilea jucător).

Exemplu: considerăm $n = 7$ (tabla are 7 coloane, 16 de poziții pe care se pot pune bile) și specificăm mutările jucătorilor prin enumerarea coloanelor pe care se pun bilele. Dacă aceste coloane sunt: 2, 4, 1, 4, 2, 5, 3, 3, 3, 4, 7, 4, 6, 5, 5, 6 atunci configurația tablei pe parcursul jocului arată ca mai sus (figura din dreapta). Scorul

primului jucător este 3 (lungimea secvenței orizontale de culoare roșie de X), scorul celui de-al doilea jucător este 4 (lungimea secvenței verticale albastre de O), .

Cerințe:

- e. Scrieți o funcție cu numele *genereazaTablaTriunghiulara* care creează un tablou bidimensional triunghiular (tabla de joc) ca mai sus cu *n* coloane și un număr de linii corespunzător. **(1 punct)**
- f. Scrieți o funcție cu numele *calculeazaConfiguratieTablaTriunghiulara* care determină și afișează pe ecran configurația tablei de joc pătrate pe baza unui fișier de intrare de forma *tablaTriunghiulara_n.txt*. **(1 punct)**
- g. Scrieți o funcție cu numele *calculeazaScoruriTablaTriunghiulara* care calculează și returnează scorurile celor doi jucatori pe baza configurației tablei triunghiulare de joc. **(0.5 puncte)**
- h. Scrieți un program care prin apeluri utile ale funcțiilor de la punctele e-g determină scorurile celor doi jucatori pentru fișierul *tablaTriunghiulara_7.txt* din arhivă. **(0.5 puncte)**

NOTĂ:

- 1. Toate datele de intrare, din toate fișierele sunt corecte. Nu trebuie să verificați corectitudinea lor.
- 2. Rezolvările corecte care nu respectă restricțiile indicate (rezolvarea unor cerințe fără a folosi funcții, utilizarea unor tablouri alocate static, utilizarea de variabile globale) vor primi punctaje parțiale.
- 3. Variabilele care se referă la tabla de joc, scorul fiecărui jucător, etc. nu trebuie să fie definite global. Ele trebuie transmise funcțiilor ca parametri. Definirea lor ca variabile globale atrage automat scăderea punctajului lucrării.
- 4. Cerințele trebuie să fie rezolvate utilizând strict limbajul C standard, ci nu limbajul C++!
- 5. Se acordă 1 punct din oficiu.