

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Пловдив, 8 – 10 юни 2018 г.

Група А, 11-12 клас

Задача А3. ХОРИЗОНТАЛНО-ВЕРТИКАЛНА ИГРА

Хвърчилко и Вятърко играят следната игра.

Игралното поле има формата на квадратна решетка от точки с размери $N \times N$, където N е цяло число, не по-малко от 2. Върху една от точките, определена по номер на ред R и номер на колонка C (номерата са от 1 до N), е поставена пионка.

Първи играе Хвърчилко, след което двамата се редуват. На всеки свой ход Хвърчилко трябва да премести пионката хоризонтално от текущото място до някоя *друга* точка наляво или надясно (без, разбира се, да напуска очертанятия на игралното поле). По пътя си пионката, евентуално, преминава през други точки. Аналогично, Вятърко пък на всеки свой ход трябва да мести пионката вертикално нагоре или надолу. **Всички точки, върху които е заставала или през които е преминавала пионката, вече са посетени и пионката не може повече да застава върху тях или да преминава през тях.**

Губи този, който не може да направи ход.

Задача

Напишете функция `play(int&, int&)`, която ще се компилира с програма на журито и ще играе срещу нея. Ще имате един инструмент за комуникация – функция `xbgame` за получаване на параметрите на играта: брой редове/стълбове и текущи координати на пионката. Чрез `play` ще правите своите ходове срещу програмата на журито. Специален изход на `play` позволява да изберете като кого ще играете: като Хвърчилко или като Вятърко. Ако играта завърши с Ваша победа, Вашата функция `play` ще получи предвидения за теста брой точки.

Детайли по реализацията

Функция `xbgame` има следния прототип:

```
void xbgame (int &N, int &R, int &C);
```

При всяко обръщение към нея Вашата функция `play` ще получава стойностите на трите параметъра, които определят играта, а именно (в този ред): размер на квадрата в брой редове/колонки с точки (тази стойност има смисъл само при първо повикване, след това остава една и съща), текущи номер на ред и номер на колонка на пионката.

Вашата функция `play` трябва да има следния прототип:

```
void play(int &R, int &C);
```

С тази функция Вие играете следващия си ход, съобщавайки на противника съответно реда и колонката на позицията, на която искате да преместите пионката. Ако искате да играете втори (т. е., като Вятърко), на първия ход върнете нули във входно-изходните параметри R и C .

Към системата изпратете Ваш файл `xbgame.cpp`. В него, освен функцията `play`, може да има всякакви инструменти, необходими за решаването на задачата. Той само не трябва да съдържа `main` и трябва да включва библиотеката `xbgame.h` чрез указание към предпроцесора

```
#include "xbgame.h"
```

в началото си.

Ограничения

$2 \leq N \leq 100, 1 \leq R, C \leq N$

В първия тестов набор, за 10% от точките, $N \leq 10$.

Във втория тестов набор, за 20% от точките, началното положение на пионката е в някой от ъглите на решетката.

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Пловдив, 8 – 10 юни 2018 г.

Група А, 11-12 клас

В третия тестов набор, за 30% от точките, началното положение на пионката е в средата на мрежата (точно в центъра за нечетно N и една от четирите централни за четно N).

В четвъртия тестов набор, за 40% от точките, няма допълнителни ограничения.

Оценяване

Всеки от четирите набора с тестове съдържа по десет игри при съответните ограничения. Всеки от четирите теста се оценява отделно: Вашата функция *play* ще получи предвидения брой точки, ако играта, описана в него, завърши с Ваша победа.

Визуализатор

За по-лесно вникване в логиката на играта, журито Ви предоставя файла `XVgame.html`, в който можете да играете в два варианта:

- Да изпълнявате ходовете и на двамата играчи.
- Да играете срещу програмата. В този случай, разбира се, нейните отговори са на случаен принцип.

Локално тестване

За локално тестване журито Ви предоставя програмния файл `Lgrader.cpp`. Запишете го в папката, в която създавате Вашето решение `xbgame.cpp`, и можете да тествате технически работата на Вашата функция *play*, като компилирате и стартирате този `Lgrader.cpp`. От стандартния вход се въвежда *един ред, съдържащ само трите цели положителни числа N , R и C , в тази последователност, разделени с интервал*. Значението им е описано по-горе. Както може да се очаква, ходовете, които прави `Lgrader` от страна на журито, не се подчиняват на никаква стратегия. Можете да промените `Lgrader.cpp` както Ви е удобно. Журито е предвидило елементарен режим на трасиране: за целта можете да премахнете знака за коментар в предпроцесорната инструкция `#define trace` в началото на файла.

Пример

Журито Ви предоставя примерен файл `Jxbgame.cpp`, който реализира функция *play*, играеща на случаен принцип. Ако го преименувате като `xbgame.cpp` или в `Lgrader.cpp` замените `#include "xbgame.cpp"` с `#include "Jxbgame.cpp"`, след компилиране на `Lgrader.cpp` ще получите работеща програма. Както очаквате, обаче, тя няма да играе много умно – това е задача на Вашия `xbgame.cpp`.