

XXXIV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 16 – 18 март 2018 г.

Група АВ, 9 – 12 клас, Ден 1

Задача АВ2. Ловец

Х и У играят следната игра върху таблица $N \times M$ (N реда на M стълба). Горният ляв ъгъл на таблицата има координати $(0,0)$. Първоначално в някои клетки на таблицата има капани. **Винаги Х започва първи**. Играчите се редуват да правят ходове с пионка (ловец) по следния начин:

1) На първия ход Х поставя ловеца в някоя клетка.

2) На всеки следващ ход играчът, който е на ход, премества ловеца в клетка, съседна по страна на тази, в която се намира **и в която няма капан**. При това преместване, ловецът поставя капан в клетката, в която е бил до момента (по този начин, капани се появяват във всички клетки, на които ловецът е стъпвал).

Губи играчът, който не може да направи ход.

Задача

Напишете функция `play()`, която ще се компилира с програма на журито и ще играе срещу нея. Тази функция ще получи таблица с първоначално разположение на капани. Тя трябва да избере дали да играе от името на Х или У и да се постарее да спечели играта. Ако спечели, то получава точките за съответния тест; ако загуби, получава 0 точки за съответния тест.

Детайли по реализацията

Функцията `play()` трябва да има следния формат:

```
void play(int n, int m, const vector<vector<char>> &table);
```

Тя се вика веднъж от програмата на журито с аргументи N и M - размерите на таблицата и двумерен масив `table`, задаващ първоначалното разположение на капаните: `table[i][j] = '#'`, когато на позиция i, j има капан и `table[i][j] = '.'`, когато клетката е празна.

За комуникация с програмата на журито Ви е предоставена функцията:

```
pair<int, int> makeMove(int i, int j);
```

Трябва да викате тази функция, за да играете поредния си ход (преместване на ловеца на позиция i, j). Ако ходът Ви е валиден (в рамките на таблицата и на съседна клетка, в която няма капан), функцията връща позицията, на която журито премества пионката (това задължително е валиден ход) или двойката числа $(-1, -1)$, ако сте победили и програмата на журито „се предава“. В този случай функцията `play()` трябва да прекрати изпълнението си. Ако направите невалиден ход или програмата на журито победи, то тя прекратява изпълнението на грейдъра и получавате 0 точки за този тест.

Ако искате първият ход да е на журито, извикайте в началото `makeMove(-1, -1)`.

Вие трябва да предадете към системата файл **hunter.cpp**, който съдържа функцията `play()`. Той може да съдържа и друг код, необходим за работата на функцията `play`, но не трябва да съдържа `main()`.

XXXIV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Национален кръг
Стара Загора, 16 – 18 март 2018 г.
Група АВ, 9 – 12 клас, Ден 1

В началото си Вашият файл трябва да съдържа `#include "hunter.h"`.

Примерна комуникация с програмата на журито:

<i>Текущо състояние на терена</i>	<i>Ход на програмата</i>	<i>Върнат резултат от журито</i>	<i>Пояснение</i>
...	makeMove(1, 1)	(1, 2)	Програмата ви играе първия си ход в средната клетка и като отговор журито играе вдясно.
... . # * ...	makeMove(0, 2)	(0, 1)	Ходът на програмата е нагоре, а на журито наляво.
. * # . # # ...	makeMove(0, 0)	(1, 0)	Ходът на програмата е наляво, отговорът на журито е надолу.
### * ## ...	makeMove(2, 0)	(2, 1)	Надолу и надясно.
### ### #* .	makeMove(2, 2)	(-1, -1)	Програмата играе надясно и това не оставя възможен ход на журито и журито се предава като връща резултат (-1, -1). При това положение функцията play трябва да приключи изпълнение.

Забележка: С '*' се отбелязва текущата позиция на ловеца.

Ограничения

$$1 \leq N, M \leq 100$$

XXXIV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг

Стара Загора, 16 – 18 март 2018 г.

Група АВ, 9 – 12 клас, Ден 1

Подзадачи:

Подзадача 1 (10 точки): $N=1$

Подзадача 2 (20 точки): Първоначално таблицата е празна (без нито един капан)

Подзадача 3 (30 точки): $N \leq 10$

Подзадача 4 (40 точки): без допълнителни ограничения

Точките за дадена подзадача се получават, ако спечелите всички игри от тестовете, предвидени за тази подзадача.

Локално тестване

За локално тестване е предоставен файл **Lgrader.cpp**. Сложете го в същата папка, в която е Вашият файл `hunter.cpp` и го компилирайте. Ще получите програма, която Ви позволява да въведете начална таблица и после да въвеждате ходовете от името на програмата на журито, докато програмата Ви се тества.

Входът на локалния грейдър е: от първия ред се въвеждат N и M – размерите на таблицата. Следват N реда, всеки от които съдържа по M символа, задаващи първоначалното разположение на капаните - '#' означава, че в съответната клетка има капан, а '.' - че няма.

Предаване на тестове към системата

Можете да предавате тестове към системата. Форматът на входните данни е същия както за локалния грейдър. Ако на изход получите число, по-голямо от 0, то Вашата програма е спечелила; ако получите 0, то е спечелила програмата на журито.