Точки

ЗАДАЧА

Разглеждаме игра, наречена Точки, при която двама играчи последователно правят ходове от една позиция към друга. Позициите имат последователни номера 1,2,..., N и са свързани със стрелки. Чрез всяка стрелка се отива от една позиция в друга. Всяка позиция се притежава от някой от двамата играчи, когото наричаме собственик на позицията. Освен това, всяка от позициите има положителна стойност. Стойностите на различните позиции са различни.

Позиция 1 се смята за начална. В началото всеки от играчите има по 0 точки.

Играта се играе както следва. За всеки ход една от позициите е текуща – да я означим с C. Преди първия ход на играта текуща е началната позиция. На всеки ход се изпълняват следните операции:

1. Ако стойността в C е по-голяма от броя на точките на собственика на позицията, неговите точки стават равни на стойността в C. В противен случай точките на собственика не се променят. И в двата случая точките на другия играч остават без промяна.
2. След това собственикът на C избира една от стрелките, излизащи от C и чрез тази стрелка отива в нова текуща позиция.

Играта завършва, когато след някой ход текущата позиция C отново е началната. Победител е играчът, който има повече точки в края на играта.

Стрелките са зададени така, че:

- Винаги има стрелка излизаща от текущата позиция.
- Всяка позиция P е достижима от началната, т.е., съществува последователност от стрелки водеща от позиция 1 до P.
- Играта винаги може да бъде доведена до край.

Напишете програма, която играе играта и печели. Игрите, в които програмата ще участва по време на тестването са такива, че могат да бъдат спечелени от нея, независимо дали ще играе като първи или като втори играч. По време на тестването противникът на програмата играе оптимално, т.е. ако програмата му даде шанс, той печели играта, а тя губи.

ВХОД И ИЗХОД

Програмата трябва да чете от стандартния вход и да пише на стандартния изход. Програмата е Играч 1, а противникът – Играч 2. Когато програмата започне работа, първо трябва да прочете следните входни данни.

От първия ред – едно цяло число: броят на позициите N, $1 \le N \le 1000$. Всеки от следващите N реда съдържа по N числа, задаващи положението на стрелките. Ако съществува стрелка от позиция с номер i към позиция с номер j, тогава j-тото число в i-тия ред е 1, а в противен случай - 0.

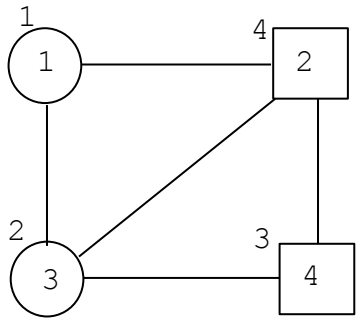
Следващият ред съдържа N цели числа: собствениците на отделните позиции. Ако позицията с номер i е собственост на Играч 1, i-тото число е 1, а в противен случай – 2.

Последният ред съдържа N цели: стойностите на отделните позиции. Ако i-тото число е j, стойността на позицията с номер i е j. За стойностите j на отделните позиции е в сила $1 \le j \le N$ и всички стойности са различни.

По време на играта програмата трябва да прави следното:

- Ако е на ход, извежда на стандартния изход номера на избраната от нея следваща текуща позиция P, $1 \le P \le N$.
- Ако на ход е противникът, тя въвежда от стандартния вход номера на избраната от него следваща текуща позиция P, $1 \le P \le N$.

Да разгледаме следния пример. Играта е показана на Фигура 1. Позициите, показани с кръг, принадлежат на Играч 1, а тези с квадрат – на Играч 2. Стойността на всяка позиция е зададена в съответния кръг или квадрат, а номерът на позицията – до кръга или квадрата. Развитието на играта е показано по-долу.



Фигура 1

stdin	stdout	обяснение
4		N
0 1 0 0		Данни за стрелките
0 0 1 1		Данни за стрелките
0 0 0 1		Данни за стрелките
1 0 0 0		Данни за стрелките
1 1 2 2		Собственост на позициите
1 3 4 2		Стойности на позициите
	2	Ход на Играч 1.
	4	Ход на Играч 1.
1		Играч 2 отива в началната позиция и играта завършва.

В края на играта Играч 1 има 3, а Играч 2 има 2 точки. Играч 1 печели.

ИНСТРУКЦИИ ЗА ПРОГРАМИСТА

В примера по-долу, `target` е цяла променлива съдържаща номер на позиция.

Ако програмирате на C++ с `iostreams`, използвайте следния фрагмент за четене от стандартния вход и писане на стандартния изход:

```
cin>>target;
cout<<target<<endl<<flush;
```

Ако програмирате на C или C++ и със `scanf` и `printf`, използвайте следния фрагмент за четене от стандартния вход и писане на стандартния изход:

```
scanf ("%d", &target);
printf("%d\n",target); fflush (stdout);
```

Ако програмирате на Pascal, използвайте следния фрагмент за четене от стандартния вход и писане на стандартния изход:

```
Readln(target);
Writeln(target);
```

ИНСТРУМЕНТИ

Ще получите програма (`score2` в Linux, `score2.exe` в Windows), която прочита описанието на играта от файл `score.in` във формата, описан по-горе. Програмата ще запише прочетеното, без да го изменя, на стандартния изход. Този изход може да бъде използван като вход от вашата програма за нуждите на тестването. След това `score2` играе със случайна стратегия, четейки ходовете на Вашата програма от стандартния си вход и записвайки своите ходове на стандартния си изход.

ОЦЕНЯВАНЕ

За всеки тестов пример, ако програмата Ви спечели, получавате пълния брой точки, а ако загуби – 0 точки. По време на проверката програмата Ви ще се изпълни два пъти. Първият път ще трябва да играе с друга програма, като ограничението на времето ще е с 1 секунда повече от обявеното, а входът и изходът и ще се записват във файл. Вторият път програмата ще бъде изпълнена като входа и се пренасочва от файл и ограничението на времето вече е каквото е обявено. Резултатите, които програмата извежда при двете изпълнения, трябва да са еднакви.