



神経衰弱 (Memory2)

表に 0 以上 $N-1$ 以下の整数が 1 つ書かれたカードが 2 枚ずつある。あなたと JOI 君は、これら $2N$ 枚のカードを用いて神経衰弱というゲームの練習をしている。

ゲームの練習を始める時点では、カードは裏向きの状態でテーブルに横一列に並べられている。左から $i+1$ 枚目 ($0 \leq i \leq 2N-1$) のカードを、カード i と呼ぶ。カード i の表に書かれた整数を A_i ($0 \leq i \leq 2N-1$) とする。最初、JOI 君とあなたには、 A_i ($0 \leq i \leq 2N-1$) がどのような値であるかは分からない。

あなたと JOI 君は、以下のやりとりを K 回まで繰り返すことができる。

1. あなたは、 $2N$ 枚のカードのうちの 2 枚のカードを指定する。
2. JOI 君は、指定された 2 枚のカードをめくり、表に書かれた整数をあなたに見えないようにこっそり見る。もし、 2 枚のカードの表に書かれた整数が等しい場合は、その値を覚えてあなたに伝える。そうでない場合は、表に書かれている整数のうち JOI 君が覚えやすい方の整数を覚えてあなたに伝える。

JOI 君にとっての整数の覚えやすさは、 N 個の整数 P_0, P_1, \dots, P_{N-1} で表される。これらの整数は、以下の 2 つの条件を満たす。

- $0 \leq P_i \leq N-1$ ($0 \leq i \leq N-1$).
- $P_i \neq P_j$ ($0 \leq i < j \leq N-1$).

JOI 君にとって i が j よりも覚えやすいことは、 $P_i < P_j$ が成り立つことと同値である。

あなたの課題は、JOI 君と K 回以下のやりとりを行うことで、それぞれのカードに書かれた整数を特定することである。ただし、あなたは、JOI 君にとっての整数の覚えやすさを表す整数 P_0, P_1, \dots, P_{N-1} がどのような値であるかを知らない。

課題

JOI 君とやりとりを行って、それぞれのカードに書かれた整数を特定するプログラムを作成せよ。

実装の詳細

あなたは、それぞれのカードに書かれた整数を特定する方法を実装した 1 個のプログラムを書かねばならない。プログラムは `Memory2_lib.h` をインクルードすること。

プログラムは、以下のルーチンを実装しなければならない。

- `void Solve(int T, int N)`

このルーチンは、各テストケースに対し 1 回だけ呼び出される。引数 T は小課題の番号を表し、 N はカードが $2N$ 枚あることを表す。

このルーチンは、`Flip` を呼び出すことによってカードに書かれた整数を特定し、その内容を `Answer` を呼び出すことによって答えなければならない。



プログラム中では以下の関数を呼び出すことができる。

- `int Flip(int I, int J)`

この関数は、JOI 君にカードを指定する際に呼び出す。引数 I, J は、JOI 君がめくるカードの番号 I, J である。

I と J はともに 0 以上 $2N - 1$ 以下の互いに異なる整数でなければならない。これを満たさない引数とともに `Flip` を呼び出した場合は **不正解 [1]** となる。

この関数は、整数 A_I と整数 A_J が等しい場合はその値を、そうでない場合は整数 A_I と整数 A_J のうち JOI 君にとって覚えやすい方の値を返す。

この関数を K 回を超えて呼び出した場合は **不正解 [2]** となる。

- `void Answer(int I, int J, int X)`

この関数は、表に整数 X が書かれたカードの番号を特定できたことを表す。

引数 I, J, X は、以下の条件を満たしていなければならない。

- $0 \leq I \leq 2N - 1$.
- $0 \leq J \leq 2N - 1$.
- $I \neq J$.
- $A_I = A_J = X$.

これらを満たさない引数とともに `Answer` を呼び出した場合は **不正解 [3]** となる。

引数 X は、以前のどの呼び出しにおける引数 X とも異ならなければならない。これが満たされない場合は **不正解 [4]** となる。

この関数は、ちょうど N 回呼び出さなければならない。これが満たされない場合は **不正解 [5]** となる。

内部での使用のために他のルーチンを実装したり、グローバル変数を宣言したりするのは自由である。ただし、あなたの提出は標準入力・標準出力、あるいは他のファイルといかなる方法でもやりとりしてはならない。

コンパイル・実行の方法

作成したプログラムをテストするための、採点プログラムのサンプルが、コンテストサイトからダウンロードできるアーカイブの中に含まれている。このアーカイブには、提出しなければならないファイルのサンプルも含まれている。

採点プログラムのサンプルは 1 つのファイルからなる。そのファイルは `grader.c` または `grader.cpp` である。作成したプログラムをテストするには、次のようにコマンドを実行する。

- C の場合

```
gcc -std=c11 -O2 -o grader grader.c Memory2.c -lm
```



- C++ の場合

```
g++ -std=c++11 -O2 -o grader grader.cpp Memory2.cpp
```

コンパイルが成功すれば、**grader** という実行ファイルが生成される。

実際の採点プログラムは、採点プログラムのサンプルとは異なることに注意すること。採点プログラムのサンプルは単一のプロセスとして起動する。このプログラムは、標準入力から入力を読み込み、標準出力に結果を出力する。

採点プログラムのサンプルの入力

採点プログラムのサンプルは標準入力から以下のデータを読み込む。

- 1 行目には、整数 T, N, K が空白を区切りとして書かれている。これらは、小課題の番号が T で、カードが $2N$ 枚あり、JOI 君とのやりとりが K 回まで許されていることを表す。
- 2 行目には、整数 P_0, P_1, \dots, P_{N-1} が空白を区切りとして書かれている。これらは JOI 君にとっての整数の覚えやすさを表す。
- 3 行目には、整数 $A_0, A_1, \dots, A_{2N-1}$ が空白を区切りとして書かれている。これらはカードに書かれた整数を表す。

採点プログラムのサンプルの出力

プログラムの実行が正常に終了した場合、採点プログラムのサンプルは標準出力へ以下の情報を 1 行で出力する (引用符は実際には出力されない)。

- 正解の場合、“Accepted” と出力する。
- 不正解の場合、不正解の種類を “Wrong Answer [2]” のように出力する。

制限

すべての入力データは以下の条件を満たす。

- $1 \leq N \leq 50$.
- $0 \leq P_i \leq N-1$ ($0 \leq i \leq N-1$).
- $P_i \neq P_j$ ($0 \leq i < j \leq N-1$).
- $0 \leq A_i \leq N-1$ ($0 \leq i \leq 2N-1$).
- どの x ($0 \leq x \leq N-1$) に対しても、 $A_i = x$ を満たす i ($0 \leq i \leq 2N-1$) はちょうど 2 つある。



小課題

小課題 1 [10 点]

以下の条件を満たす.

- $T = 1$.
- $K = 10\,000$.
- $P_i = i$ ($0 \leq i \leq N - 1$).

小課題 2 [50 点]

以下の条件を満たす.

- $T = 2$.
- $K = 400$.
- $P_i = i$ ($0 \leq i \leq N - 1$).

小課題 3 [40 点]

以下の条件を満たす.

- $T = 3$.
- $K = 300$.

やりとりの例

採点プログラムのサンプルが読み込む入力の例と、それに対応するルーチンの呼び出しの例を以下に示す.

入力例	ルーチンの呼び出しの例	
	呼び出し	戻り値
1 3 10000 0 1 2 1 0 2 0 1 2	Flip(0, 2)	1
	Flip(0, 4)	1
	Flip(1, 2)	0
	Answer(0, 4, 1)	
	Flip(1, 3)	0
	Flip(5, 2)	2
	Flip(4, 5)	1
	Answer(1, 3, 0)	
	Answer(5, 2, 2)	

この例での関数の呼び出しは、必ずしも意味のある呼び出しとは限らないことに注意せよ.