



マトリョーシカ人形 (Matryoshka)

あなたは、マトリョーシカ人形を販売する店を開こうとしている。そこで、あなたは N 個のマトリョーシカ人形を工場に注文した。これらには 1 から N までの番号が付けられている。このうち i 番目 ($1 \leq i \leq N$) のマトリョーシカ人形は、底面の直径 R_i cm で高さ H_i cm の、中が空洞の直円柱とみなすことができる。

マトリョーシカ人形は入れ子にして保管することができる。それぞれのマトリョーシカ人形は、底面の直径と高さがともにより小さい他のマトリョーシカ人形を 1 つだけ収納することが出来る。収納されるマトリョーシカ人形は、他のマトリョーシカ人形を収納していてもよい。

ある日、マトリョーシカ人形を注文した工場から連絡が届いた。注文した N 個のマトリョーシカ人形はすべてを一度に用意することはできないので、 N 個のマトリョーシカ人形のうち底面の直径が A cm 以上であり、高さが B cm 以下であるものすべてが事前に届くそうだ。

A, B の値は急に変更されるかもしれない。そこで、あなたは、 Q 個の組 (A_j, B_j) ($1 \leq j \leq Q$) のそれぞれに対して、事前に届くマトリョーシカ人形を入れ子にして保管したときの、どのマトリョーシカ人形にも収納されていないマトリョーシカ人形の個数の最小値をあらかじめ求めておくことにした。

課題

それぞれのマトリョーシカ人形の底面の直径と高さの情報と、 Q 個の組 (A_j, B_j) ($1 \leq j \leq Q$) が与えられる。それぞれの組について、事前に届くマトリョーシカ人形を入れ子にして保管したときの、どのマトリョーシカ人形にも収納されていないマトリョーシカ人形の個数の最小値を求めるプログラムを作成せよ。

入力

標準入力から以下のデータを読み込め。

- 1 行目には、整数 N, Q が空白を区切りとして書かれている。これは、注文したマトリョーシカ人形の個数が N 個であり、 A, B の値の組が Q 個与えられることを表す。
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には、整数 R_i, H_i が空白を区切りとして書かれている。これは、 i 番目のマトリョーシカ人形は、底面の直径が R_i cm で高さが H_i cm であることを表す。
- 続く Q 行のうちの j 行目 ($1 \leq j \leq Q$) には、整数 A_j, B_j が空白を区切りとして書かれている。

出力

出力は Q 行からなる。 j 行目 ($1 \leq j \leq Q$) には、組 (A_j, B_j) について、事前に届くマトリョーシカ人形を入れ子にして保管したときの、どのマトリョーシカ人形にも収納されていないマトリョーシカ人形の個数の最小値を出力せよ。



制限

すべての入力データは以下の条件を満たす。

- $1 \leq N \leq 200\,000$.
- $1 \leq Q \leq 200\,000$.
- $1 \leq R_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).
- $1 \leq H_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).
- $1 \leq A_j \leq 1\,000\,000\,000$ ($1 \leq j \leq Q$).
- $1 \leq B_j \leq 1\,000\,000\,000$ ($1 \leq j \leq Q$).

小課題

小課題 1 [11 点]

以下の条件を満たす。

- $N \leq 10$.
- $Q = 1$.

小課題 2 [15 点]

以下の条件を満たす。

- $N \leq 100$.
- $Q = 1$.

小課題 3 [25 点]

以下の条件を満たす。

- $N \leq 2\,000$.
- $Q \leq 2\,000$.

小課題 4 [49 点]

追加の制約はない。



入出力例

入力例 1	出力例 1
7 3	0
9 5	1
3 7	2
10 6	
5 10	
2 6	
10 10	
4 1	
10 5	
3 5	
3 9	

- $(A, B) = (10, 5)$ のとき，底面の直径が 10 cm 以上で高さが 5 cm 以下であるマトリョーシカ人形は 1 つも存在しないので，0 を出力する．
- $(A, B) = (3, 5)$ のとき，底面の直径が 3 cm 以上で高さが 5 cm 以下であるマトリョーシカ人形，すなわち 1 番目，7 番目のマトリョーシカ人形が届く．7 番目のマトリョーシカ人形は 1 番目のマトリョーシカ人形の中に収納することができる．どのマトリョーシカ人形にも収納されていないマトリョーシカ人形の個数の最小値は 1 である．
- $(A, B) = (3, 9)$ のとき，底面の直径が 3 cm 以上で高さが 9 cm 以下であるマトリョーシカ人形，すなわち 1 番目，2 番目，3 番目，7 番目のマトリョーシカ人形が届く．この場合，7 番目のマトリョーシカ人形を 1 番目のマトリョーシカ人形の中に収納して，1 番目のマトリョーシカ人形を 3 番目のマトリョーシカ人形の中に収納することができる．どのマトリョーシカ人形にも収納されていないマトリョーシカ人形の個数の最小値は 2 である．



入力例 2	出力例 2
10 8	3
14 19	1
9 16	3
11 2	5
7 18	0
20 16	2
9 5	1
10 9	3
20 6	
4 17	
13 8	
7 14	
9 3	
9 13	
4 19	
12 4	
19 16	
18 10	
7 14	



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	Flip(0, 2)	1
0 1 2	Flip(0, 4)	1
1 0 2 0 1 2	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.



ソリティア (Solitaire)

JOI 君は、縦に 3 マス、横に N マスの方眼状のボードと、いくつかのコマを使ってゲームをしている。ゲームの初期状態において、1 個以上のマスにはコマが置かれており、また、1 個以上のマスにはコマが置かれていない。

このゲームの目的は、コマの置かれていないマスにコマを 1 個ずつ置いていくことで、ボード上のすべてのマスにコマが置かれた状態にすることである。ただし、あるマスにコマを置ける条件は、以下のいずれかが満たされることである。

- そのマスの一つ上のマスと一つ下のマスの両方にコマが置かれている。
- そのマスの一つ左のマスと一つ右のマスの両方にコマが置かれている。

JOI 君はゲームの初期状態から始めて、目的を達成するまでにコマを置いていく順番が全部で何通りあるのかが気になった。ただしこの値は非常に大きくなることがある。

あなたの課題は、JOI 君の代わりに、ゲームの初期状態から目的を達成するまでにコマを置いていく順番の個数を 1 000 000 007 で割った余りを求めることである。

課題

ゲームの初期状態が与えられたとき、目的を達成するまでにコマを置いていく順番の個数を 1 000 000 007 で割った余りを求めるプログラムを作成せよ。

入力

標準入力から以下のデータを読み込め。

- 1 行目には、整数 N が書かれている。これは、ゲームで使うボードの大きさが、縦に 3 マス、横に N マスであることを表す。
- 続く 3 行のそれぞれには、 N 文字からなる文字列が書かれている。各文字は 'o' もしくは 'x' である。この 3 行のうちの i 行目 ($1 \leq i \leq 3$) の左から j 文字目 ($1 \leq j \leq N$) は、ボードの上から i 行目、左から j 列目のマスの初期状態を表す。この文字が 'o' のときは、ゲームの初期状態においてそのマスにコマが置かれていることを表す。また、'x' のときは、ゲームの初期状態においてそのマスにコマが置かれていないことを表す。

出力

標準出力に、目的を達成するまでにコマを置いていく順番の個数を 1 000 000 007 で割った余りを 1 行で出力せよ。



制限

すべての入力データは以下の条件を満たす。

- $1 \leq N \leq 2000$.

小課題

小課題 1 [10 点]

以下の条件を満たす。

- ゲームの初期状態において、コマの置かれていないマス個数は 16 個以下である。
- $N \leq 30$.

小課題 2 [12 点]

- ゲームの初期状態において、コマの置かれていない各マスについて、そのマスに上下左右に隣り合うマスのうちコマの置かれていないマス個数が 2 個以下である。

小課題 3 [20 点]

以下の条件を満たす。

- ゲームの初期状態において、コマの置かれていないマスが縦に 3 個連続して並ぶことはない。
- $N \leq 30$.

小課題 4 [38 点]

- $N \leq 300$ を満たす。

小課題 5 [20 点]

追加の制約はない。



入出力例

入力例 1	出力例 1
3 OXO XXO OXO	14

この入力例では、ゲームの初期状態は以下の通りである (コマの置かれたマスを○で表す)。

○		○
		○
○		○

次のいずれかの表に従ってコマを置いていくことで、目的を達成することができる (番号はコマを置いていく順番を表す)。

○ 1 ○ 2 3 ○ ○ 4 ○	○ 1 ○ 2 4 ○ ○ 3 ○	○ 1 ○ 3 4 ○ ○ 2 ○	○ 1 ○ 4 3 ○ ○ 2 ○	○ 2 ○ 1 3 ○ ○ 4 ○
○ 2 ○ 1 4 ○ ○ 3 ○	○ 3 ○ 1 2 ○ ○ 4 ○	○ 4 ○ 1 2 ○ ○ 3 ○	○ 3 ○ 1 4 ○ ○ 2 ○	○ 4 ○ 1 3 ○ ○ 2 ○
○ 2 ○ 3 4 ○ ○ 1 ○	○ 2 ○ 4 3 ○ ○ 1 ○	○ 3 ○ 2 4 ○ ○ 1 ○	○ 4 ○ 2 3 ○ ○ 1 ○	

目的を達成する順番はこれらの 14 通りのみであるので、14 を出力する。

入力例 2	出力例 2
10 OOXOOXOXOO XOOXXXOXXX OXOXOOOOOO	149022720

入力例 2 はすべての小課題の制約を満たす。



入力例 3	出力例 3
10 00X0XX0X00 OXXXXX0XXX OX00X0X0X0	0

ゲームの初期状態によっては、目的を達成できないこともある。

入力例 4	出力例 4
20 OX00X0X00X0X00X0X0X0 OXXXX0XXXX00XXXXXX00X OX00X0X00X00X00X0X00	228518545