

Problem A. Matryoshka

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

당신은 마트료시카 인형을 판매하는 가게를 열기로 했다. 당신은 N 개의 마트료시카 인형을 공장에 주문했다. 여기에는 1부터 N 까지의 번호가 붙어 있었다. 그 중 i 번째($1 \leq i \leq N$) 마트료시카 인형은 바닥의 지름이 R_i cm, 높이가 H_i cm인 속이 빈 원기둥으로 생각할 수 있다.

마트료시카 인형은 중첩해서 안에 넣을 수 있다. 각각의 마트료시카 인형은, 자신보다 지름과 높이가 모두 작은 마트료시카 인형을 1개까지만 안에 넣을 수 있다. 수납된 마트료시카 인형 안에 다른 인형을 넣을 수도 있다.

어느 날, 마트료시카 인형을 주문한 공장에서 연락이 왔다. 주문한 N 개의 마트료시카 인형을 모두 한번에 준비할 수 없어서, N 개의 마트료시카 인형 중 **바닥의 직경이 A cm 이상, 높이 B cm 이하**의 마트료시카만 모두 사전에 보내준다고 한다.

A 와 B 는 갑자기 변할지도 모른다. 그래서 당신은, Q 개의 쌍 (A_j, B_j) ($1 \leq j \leq Q$) 각각에 대해, 사전에 도착하는 마트료시카 인형을 중첩해서 넣었을 때, 어떤 마트료시카 인형에도 들어가지 못하는 인형의 갯수의 최솟값을 구하기로 하였다.

각각의 마트료시카 인형의 바닥의 지름과 높이에 대한 정보와, Q 개의 쌍 (A_j, B_j) ($1 \leq j \leq Q$)이 주어진다. 각각의 쌍에 대해서, 사전에 도착하는 마트료시카 인형을 중첩해서 넣을 때, 어떤 마트료시카 인형에도 들어가지 못하는 마트료시카 인형의 갯수의 최솟값을 구하는 프로그램을 작성하여라.

Input

첫째 줄에는, 정수 N, Q 가 공백으로 구분되어 들어온다. 이것은, 주문한 마트료시카 인형이 N 개이고, (A, B) 쌍의 수가 Q 개라는 것을 의미한다.

다음 N 개의 줄의 i 번째 줄($1 \leq i \leq N$)에는, 정수 R_i, H_i 가 공백으로 구분되어 들어온다. 이것은, i 번째 마트료시카 인형은, 바닥의 지름이 R_i cm 이고, 높이가 H_i cm 임을 의미한다.

다음 Q 개의 줄의 j 번째 줄($1 \leq j \leq Q$)에는, 정수 A_j, B_j 가 공백으로 구분되어 들어온다.

Output

출력은 Q 개의 줄로 한다. j 번째 줄($1 \leq j \leq Q$)에는, 쌍 (A_j, B_j) 에 대해, 사전에 도착한 마트료시카 인형을 중첩해서 넣을 때, 어떤 마트료시카 인형에도 들어가지 못하는 마트료시카 인형의 갯수의 최솟값을 출력하여라.

Constraints

모든 입력데이터는 다음의 조건을 만족한다.

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

Subtask 1 (11 points)

다음의 조건을 만족한다.

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

Subtask 2 (15 points)

다음의 조건을 만족한다.

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

Subtask 3 (25 points)

다음의 조건을 만족한다.

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

Subtask 4 (49 points)

추가 제한조건이 없다.

Examples

standard input	standard output
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	

- $(A, B) = (10, 5)$ 일 때, 바닥의 지름이 10cm 이상이고, 높이가 5cm 이하인 마트료시카 인형은 한 개도 없기 때문에 0을 출력한다.
- $(A, B) = (3, 5)$ 일 때, 바닥의 직경이 3cm 이상이고, 높이가 5cm 이하인 마트료시카 인형, 즉 1번째와 7번째 마트료시카 인형이 사전에 도착한다. 7번째 마트료시카 인형은 1번째 마트료시카 인형 안에 넣을 수 있다. 어떤 마트료시카 인형에도 들어가 있지 않은 마트료시카 인형의 최솟값은 1이다.
- $(A, B) = (3, 9)$ 일 때, 바닥의 직경이 3cm 이상이고, 높이가 9cm 이하인 마트료시카 인형, 즉 1번째, 2번째, 3번째와 7번째 마트료시카 인형이 사전에 도착한다. 이 때, 7번째 마트료시카 인형은 1번째 마트료시카 인형의 안에 들어갈 수 있고, 1번째 마트료시카 인형은 3번째 마트료시카 인형 안에 들어갈 수 있다. 어떤 마트료시카 인형에도 들어가 있지 않은 마트료시카 인형의 최솟값은 2이다.

standard input	standard output
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	

This page is intentionally left blank

Problem B. Memory2

Time limit: 2 seconds
Memory limit: 256 megabytes

0이상 $N - 1$ 이하의 정수가 앞면에 1개씩 쓰인 카드가 $2N$ 장씩 있다. 당신과 JOI군은, 이 $2N$ 장의 카드를 사용해서 신경쇠약이라는 게임의 연습을 하고 있다. 게임의 연습을 시작할 때에는, 카드가 모두 뒷면인 채로 책상 위에 가로 일렬로 놓여 있다. 왼쪽부터 $i + 1$ 번째 ($0 \leq i \leq 2N - 1$) 카드를, **카드 i** 라고 부른다. 카드 i 에 쓰인 정수를 A_i ($0 \leq i \leq 2N - 1$)라고 한다. 처음에, JOI군과 당신은, A_i ($0 \leq i \leq 2N - 1$)를 알지 못한다. 당신과 JOI군은, 다음의 주고받기를 K 번 까지 실행할 수 있다.

- 당신은, $2N$ 장의 카드 중 2장의 카드를 정한다.
- JOI군은, 정해진 2장의 카드를 뒤집어, 뒤에 무슨 정수가 쓰여 있는지 당신에게 보이지 않게 몰래 본다. 만약, 2장의 카드의 뒤에 쓰인 정수가 같은 경우에는, 그 수를 당신에게 알려준다. 그렇지 않은 경우에는, 표에 쓰인 정수 중 JOI군이 기억하기 쉬운 정수를 기억해서 당신에게 알려준다.

JOI군이 정수를 기억하기 쉬운 정도는, N 개의 정수 P_0, P_1, \dots, P_{N-1} 로 표시된다. 이 정수는, 다음 두가지 조건을 만족한다

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

JOI군에게, i 가 j 보다 기억하기 쉽다는 것은, $P_i < P_j$ 가 성립하는 것과 동치이다.

당신의 문제는, JOI군과 K 번 이하의 주고받기를 실행하여, 각각의 카드에 쓰인 정수를 특정하는 것이다. 단, 당신은, JOI군이 정수를 기억하기 쉬운 정도를 나타내는 정수 P_0, P_1, \dots, P_{N-1} 가 어떤 수인지 알지 못한다.

JOI군과 주고받기를 실행하여, 각각의 카드에 쓰인 수를 특정하는 프로그램을 작성하여야.

당신은, 각각의 카드에 쓰인 정수를 특정하는 방법이 구현된, 1개의 프로그램을 작성해야 한다. 프로그램은 `Memory2.lib.h`를 `include`해야한다.

Interaction Protocol

프로그램에는, 다음의 함수가 구현되어야 한다.

- `void Solve(int T, int N)` 이 함수는, 각 테스트케이스에 대해 1번만 실행된다. 인자 T 는 Subtask의 번호이고, N 은 카드가 $2N$ 개 있는 것을 의미한다.

이 함수는, `Flip`를 호출하는 것으로, 카드에 쓰인 정수를 특정하여, 그 내용을 `Answer`를 호출하는 것으로 답해야한다.

프로그램중에 이하의 함수를 호출하는 것이 가능하다.

- `int Flip(int I, int J)` 이 함수는, JOI군에게 카드를 정할 때 호출한다. 인자 I, J 는, JOI군이 뒤집을 카드의 번호인 I, J 이다.

I 와 J 는 0이상 $2N - 1$ 이하인 서로 다른 수여야 한다. 이것을 만족하지 않게 `Flip`을 호출하면 **오답 [1]**이 된다.

이 함수는, 정수 A_I 와 정수 A_J 가 같은 경우 그 수를, 아닐 경우엔 정수 A_I 와 정수 A_J 중 JOI군이 기억하기 쉬운 값을 반환한다.

이 함수를 K 번 초과하여 호출할 경우 **오답 [2]**이 된다.

- `void Answer(int I, int J, int X)` 이 함수는, 정수 X 가 쓰인 카드의 번호를 특정할 때 호출한다. 인자 I, J, X 는, 다음의 조건을 만족해야한다.

- * $0 \leq I \leq 2N - 1$
- * $0 \leq J \leq 2N - 1$

- * $I \neq J$
- * $A_I = A_J = X$

이것을 만족하지 않는 인자로 Answer를 호출한 경우, **오답 [3]**이 된다.

인자 x 는, 전에 호출되었던 인자 x 와 달라야 한다. 이것을 만족하지 않은 경우 **오답 [4]**가 된다. 이 함수는, 정확히 N 번 호출되어야 한다. 이것을 만족하지 않은 경우 **오답 [5]**가 된다.

내부 사용을 위해서 다른 함수를 구현하거나, 글로벌 변수를 선언하는것은 자유이다. 하지만, 당신의 제출은 표준 입출력이나, 다른 함수에 접근하면 안 된다.

작성한 프로그램을 테스트하기 위한 채점 프로그램 샘플이, 콘테스트 사이트에서 다운로드 받을 수 있는 아카이브 안에 있다. 이 아카이브는, 제출해야하는 파일의 샘플도 들어있다.

채점 프로그램 샘플은 1개의 파일이다. 이 파일은 `grader.c` 혹은 `grader.cpp`이다. 작성한 프로그램을 테스트 하기 위해서는, 다음의 커맨드를 실행한다.

- C의 경우 `gcc -std=c11 -O2 -o grader grader.c Memory2.c -lm`
- C++의 경우 `g++ -std=c++11 -O2 -o grader grader.cpp Memory2.cpp`

컴파일에 성공하면, `grader`라는 이름의 파일이 생성된다.

실제의 채점 프로그램은, 채점 프로그램 샘플과는 다르므로 주의한다. 채점 프로그램의 샘플은 단일 프로세스로 실행된다. 이 프로그램은, 표준입력에서 입력을 받아서, 표준출력으로 결과를 출력한다.

채점 프로그램 샘플은, 표준입력에서 다음과 같은 데이터를 읽는다.

Input

첫째 줄에는, 정수 T, N, K 가 공백으로 구분되어 들어온다. 이것은, Subtask의 번호 T 이고, 카드가 $2N$ 개 있고, JOI군과 주고받기가 K 번 까지 가능하다는 것을 의미한다.

둘째 줄에는, 정수 P_0, P_1, \dots, P_{N-1} 이 공백으로 구분되어 들어온다. 이것은 JOI군이 카드를 기억하기 쉬운 정도를 의미한다.

셋째 줄에는, 정수 $A_0, A_1, \dots, A_{2N-1}$ 이 공백으로 구분되어 들어온다. 이것은 카드에 쓰인 정수를 의미한다.

Output

프로그램의 실행이 정상적으로 종료된 경우, 채점 프로그램 샘플은 표준출력으로 다음의 정보를 첫째 줄에 출력한다. (따옴표는 출력되지 않는다.)

- 정답일 경우, "Accepted"를 출력한다.
- 오답일 경우, 오답의 종류를 "Wrong Answer [2]"처럼 출력한다.

Constraints

모든 입력데이터는 다음의 조건을 만족한다.

- $1 \leq N \leq 50$
- $1 \leq P_i \leq N - 1$ ($0 \leq i \leq N - 1$)
- $P_i \neq P_j$ ($0 \leq i < j \leq N - 1$)
- $0 \leq A_i \leq N - 1$ ($0 \leq i \leq 2N - 1$)
- 어떤 x ($0 \leq x \leq N - 1$) 에 대해서도, $A_i = x$ 를 만족하는 i ($0 \leq i \leq 2N - 1$) 가 정확히 2개 존재한다.

Subtask 1 (10 points)

다음의 조건을 만족한다.

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

Subtask 2 (50 points)

다음의 조건을 만족한다.

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

Subtask 3 (40 points)

다음의 조건을 만족한다.

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

Example

다음은 채점 프로그램 샘플이 입력받은 입력 예제와, 그에 대응되는 함수 호출 예이다.

standard input	interaction	
	Function call	Return value
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)	

이 예제의 함수 호출은, 반드시 의미 있는 호출은 아니라는 점에 주의하라.

This page is intentionally left blank

Problem C. Solitaire

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

JOI군은, 세로 3칸, 가로 N 칸의 직사각형 모양의 보드와, 몇개의 말을 가지고 게임을 하고 있다. 게임의 초기상태에는, 1개 이상에 칸에 말이 놓여져 있고, 1개 이상의 칸에 말이 놓여 있지 않다. 이 게임의 목적은, 말이 놓여 있지 않은 칸에 말을 1개씩 놓아서, 보드 위에 모든 칸에 말이 위치해 있는 상태로 만드는 것이다. 단, 어떤 칸에 말을 놓으려면 다음 중 하나의 조건을 만족해야한다.

- 그 칸의 한 칸 위 칸과 한 칸 아래 칸 양쪽에 말이 놓여 있다.
- 그 칸의 한 칸 왼 칸과 한 칸 오른 칸 양쪽에 말이 놓여 있다.

JOI군은 게임의 초기상태에서 시작하여, 목적에 맞게 말을 놓는 순서가 모두 몇 가지인지가 궁금했다. 단, 수가 매우 커 질 수 있다.

당신의 과제는, JOI군 대신, 게임의 초기상태에서 목적에 맞게 칸에 말을 놓는 순서의 가짓수를 1 000 000 007로 나눈 나머지를 구하는 것이다.

게임의 초기상태가 주어질 때, 목적에 맞게 칸에 말을 놓는 순서의 가짓수를 1 000 000 007로 나눈 나머지를 구하는 프로그램을 작성하여라.

Input

첫째 줄에는, 정수 N 이 들어온다. 이것은, 게임에 사용하는 보드의 크기가 세로 3칸, 가로 N 칸이라는 것을 의미한다.

이후 3개의 줄 각각에는, N 개의 문자로 이루어진 문자열이 들어온다. 각 문자는 'o'나 'x'이다. 이 3개의 줄 중 i 번째 ($1 \leq i \leq 3$) 줄의 왼쪽에서 j 번째 문자 ($1 \leq j \leq N$)에는, 보드의 위쪽에서 i 번째 행, 왼쪽에서 j 번째 열에 해당하는 칸의 초기상태를 의미한다. 이 문자가 'o'인 경우, 게임의 초기상태에 그 칸에는 말이 놓여 있는 것을 의미한다. 'x'인 경우, 게임의 초기상태에 그 칸에는 말이 놓여 있지 않은 것을 의미한다.

Output

목적에 달성하기 위해 말을 배치하는 순서의 가짓수를 1 000 000 007로 나눈 나머지를 첫째 줄에 출력한다.

Constraints

모든 입력데이터는 다음의 조건을 만족한다.

- $1 \leq N \leq 2\,000$

Subtask 1 (10 points)

다음의 조건을 만족한다.

- 게임에 초기상태에, 말이 위치해 있지 않은 칸의 갯수는 16개 이하이다.
- $N \leq 30$

Subtask 2 (12 points)

- 게임의 초기상태에, 말이 놓여있지 않은 각 칸에 대해, 그 칸의 상하좌우에 말이 놓여 있는 칸의 갯수가 2개 이상이다.

Subtask 3 (20 points)

다음의 조건을 만족한다.

- 게임의 초기상태에, 말이 위치해있지 않은 칸이 세로로 3개 연달아 존재하지 않는다.
- $N \leq 30$

Subtask 4 (28 points)

- $N \leq 300$ 을 만족한다.

Subtask 5 (20 points)

추가 제한조건이 없다.

Examples

standard input	standard output
3 oxo xxo oxo	14

이 입력 예제에서는, 게임의 초기상태가 다음과 같다. (말이 위치해 있는 칸을 ○로 표시한다.)

○		○
		○
○		○

다음 각각의 표에 쓰인 방법으로 말을 놓으면, 목적을 달성할 수 있다. (번호는 말을 놓는 순서를 의미한다.)

○ 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 ○	

목적을 달성하는 순서는 위의 14가지가 있으므로, 14를 출력한다.

standard input	standard output
10 ooxooxoxoo xooxxxoxxx oxoxoooooo	149022720

입력 예제 2는 모든 Subtask의 제한조건을 만족한다.

standard input	standard output
10 ooxooxoo oxxxxxxx ooxooxoo	0

게임의 초기상태에 대해서, 목적을 달성하지 못할 수도 있다.

standard input	standard output
20 ooxooxooxooxooxoo oxxxoxxxxoxxxxoxx ooxooxooxooxooxoo	228518545

This page is intentionally left blank