

Problem A. 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