

카멜레온의 사랑

JOI 동물원에는 1번부터 $2N$ 번까지의 번호가 붙어있는 $2N$ 마리의 카멜레온이 있다. 그 중, N 마리의 카멜레온은 성별이 X이다. 다른 N 마리의 카멜레온은 성별이 Y이다.

각 카멜레온에는 **원본 색**이 있다. 카멜레온의 원본 색에 대해서는 다음과 같은 사실이 알려져 있다.

- 성별 X인 카멜레온 N 마리의 원본 색은 모두 다르다.
- 성별이 X인 각 카멜레온에 대해, 원본 색이 같은 성별이 Y인 카멜레온이 유일하게 존재한다.

지금 JOI 동물원에는 새로운 사랑이 싹트는 계절이다. 각 카멜레온은 다른 카멜레온을 **사랑**한다. 카멜레온의 사랑에 대해서는 다음과 같은 사실이 알려져 있다.

- 각 카멜레온은 성별이 자신과 다른 카멜레온을 정확히 한 마리 사랑한다.
- 어떤 카멜레온과 그 카멜레온이 사랑하는 카멜레온의 원본 색은 다르다.
- 같은 카멜레온을 사랑하는 서로 다른 두 카멜레온은 존재하지 않는다.

당신은 카멜레온 몇 마리를 모아서 미팅을 주선하려고 한다. 미팅에 참석한 카멜레온 s 에 대해, s 가 카멜레온 t 를 사랑한다고 하자. s 의 **피부색**은 다음과 같이 정해진다.

- t 가 미팅에 참석하면, s 의 피부색은 t 의 원본 색이다.
- t 가 미팅에 참석하지 않으면, s 의 피부색은 s 의 원본 색이다.

카멜레온의 피부색은 어떤 미팅에 참석하냐에 따라 바뀔 수 있다. 당신이 주선하는 미팅에 대해, 미팅에 참석한 카멜레온의 피부색이 총 몇 종류인지 알 수 있다.

당신은 미팅을 20 000번 이하로 주선해서 원본 색이 같은 카멜레온 쌍을 모두 알고 싶다.

카멜레온의 수가 주어질 때, 미팅을 20 000번 이하로 주선해서 원본 색이 같은 카멜레온 쌍을 모두 정하는 프로그램을 작성하여라.

구현 명세

당신은 파일 하나를 제출해야 한다.

이 파일의 이름은 `chameleon.cpp`이다. 파일은 다음 함수를 구현해야 한다. 또한, `chameleon.h`를 include 해야 한다.

- `void Solve(int N)`
이 함수는 각 테스트 케이스마다 정확히 한 번 불린다.
 - 인자 N 은 성별이 X인 카멜레온의 수 N 을 나타낸다.

당신의 프로그램은 다음 함수를 호출 할 수 있다.

- `int Query(const std::vector<int> &p)`
당신은 이 함수를 호출함으로써 미팅을 주선할 수 있다.
 - * 인자 p 는 미팅에 참여하는 카멜레온의 목록이다.
 - * 이 함수의 반환값은 미팅에 참석한 카멜레온의 피부색이 몇 종류인지이다.
 - * p 에 있는 각 원소는 1 이상 $2N$ 이하의 정수여야 한다. 이를 만족하지 않은 경우에는 **오답 [1]**이 된다.

- * p 에 있는 각 원소는 서로 달라야 한다. 이를 만족하지 않은 경우에는 **오답 [2]**이 된다.
- * 당신은 이 함수를 20 000번 이상 호출해서는 안 된다. 이를 만족하지 않은 경우에는 **오답 [3]**이 된다.

– void Answer(int a, int b)

이 함수를 사용하여, 같은 원본 색을 가진 카멜레온의 쌍을 답할 수 있다.

- * 인자 a 와 b 는 a 번째 카멜레온과 b 번째 카멜레온의 원본 색이 같다는 것을 의미한다.
- * $1 \leq a \leq 2N$ 과 $1 \leq b \leq 2N$ 을 만족해야 한다. 이를 만족하지 않을 경우 **오답 [4]**이 된다.
- * a 와 b 로 주어진 수는 2번 이상 나타나서는 안 된다. 이를 만족하지 않은 경우에는 **오답 [5]**이 된다.
- * 서로 원본 색을 가진 카멜레온을 지정할 경우 **오답 [6]**이 된다.
- * 함수 Answer는 정확히 N 번 호출될 필요가 있다. 함수 Solve의 실행 종료 시에 함수 Answer의 호출 횟수가 N 번이 아닐 경우 **오답 [7]**이 된다.

참고 사항

- 당신의 프로그램은 내부에서 사용할 목적으로 함수나 전역변수를 사용할 수 있다.
- 당신의 프로그램은 표준 입출력을 사용해서는 안 된다. 당신의 프로그램은 어떠한 방법으로도 다른 파일에 접근해서는 안 된다. 단, 당신의 프로그램은 디버그 목적으로 표준 에러출력에 출력할 수 있다.

당신은 대회 홈페이지의 아카이브에서 프로그램을 테스트하기 위한 목적의 샘플 그레이더를 받을 수 있다. 아카이브는 당신의 프로그램의 예제 소스 또한 첨부되어 있다. 샘플 그레이더는 파일 grader.cpp이다. 당신의 프로그램을 테스트 하기 위해서, grader.cpp, chameleon.cpp, chameleon.h를 같은 디렉토리 안에 놓고, 컴파일하기 위해 다음 커맨드를 실행하여라.

- `g++ -std=gnu++14 -O2 -o grader grader.cpp chameleon.cpp`

컴파일이 성공적이면, 파일 grader가 생성된다.

실제 그레이더와 샘플 그레이더는 다름에 주의하여라. 샘플 그레이더는 하나의 프로세스에서 실행되며, 입력을 표준 입력으로부터 받고, 출력을 표준 출력에 출력한다.

입력 형식

샘플 그레이더는 표준 입력에서 다음과 같은 형식으로 입력받는다.

N
 $Y_1 \cdots Y_{2N}$
 $C_1 \cdots C_{2N}$
 $L_1 \cdots L_{2N}$
 Y_i 는 ($1 \leq i \leq 2N$) i 번째 카멜레온의 성별을 나타내고, 0 혹은 1이다. 0이면 성별이 X이고, 1이면 성별이 Y이다.
 C_i 는 ($1 \leq i \leq 2N$) i 번째 카멜레온의 원본 색을 나타내고, 1 이상 N 이하의 정수이다.
 L_i 는 ($1 \leq i \leq 2N$) i 번째 카멜레온이 사랑하는 카멜레온의 번호이다.

출력 형식

프로그램이 정상적으로 종료되었다면, 샘플 그레이더는 다음과 같은 정보를 표준 출력에 출력한다. (따옴표는 출력하지 않는다.)

- 정답으로 판단된 경우, Query함수의 호출 횟수를 “Accepted: 100”과 같은 형식으로 출력한다.

- 오답으로 판단된 경우, 오답의 종류를 “Wrong Answer [1]”과 같은 형식으로 출력한다.

프로그램이 다양한 오답의 종류에 속해 있으면, 샘플 그레이더는 그중 하나만 출력할 것이다.

제한

모든 입력 데이터는 다음의 조건을 만족한다. Y , C , L 의 의미는 입력 형식을 참고하여라.

- $2 \leq N \leq 500$.
- $0 \leq Y_i \leq 1$ ($1 \leq i \leq 2N$).
- 각 j ($1 \leq j \leq N$) 에 대해, $Y_i = 0$ 과 $C_i = j$ 를 만족하는 유일한 i 가 ($1 \leq i \leq 2N$) 존재한다.
- 각 j ($1 \leq j \leq N$) 에 대해, $Y_i = 1$ 과 $C_i = j$ 를 만족하는 유일한 i 가 ($1 \leq i \leq 2N$) 존재한다.
- $1 \leq L_i \leq 2N$ ($1 \leq i \leq 2N$).
- $Y_i \leq Y_{L_i}$ ($1 \leq i \leq 2N$).
- $C_i \leq C_{L_i}$ ($1 \leq i \leq 2N$).
- $L_k \neq L_l$ ($1 \leq k < l \leq 2N$).

서브태스크 1 (4 점)

- $L_{L_i} = i$. ($1 \leq i \leq 2N$).

서브태스크 2 (20 점)

- $N \leq 7$.

서브태스크 3 (20 점)

- $N \leq 50$.

서브태스크 4 (20 점)

- $Y_i = 0$ ($1 \leq i \leq N$).

서브태스크 5 (36 점)

추가 제한조건이 없다.

예제

이 함수는 그레이더의 예제 입력과 해당하는 함수 호출을 보여준다.

예제 입력	예제 함수 호출		
	호출	호출	반환값
4	Solve(4)		
1 0 1 0 0 1 1 0		Query([])	0
4 4 1 2 1 2 3 3		Query([6, 2])	2
4 3 8 7 6 5 2 1		Query([8, 1, 6])	2
		Query([7, 1, 3, 5, 6, 8])	4
		Query([8, 6, 4, 1, 5])	3
		Answer(6, 4)	
		Answer(7, 8)	
		Answer(2, 1)	
		Answer(3, 5)	

대회 홈페이지의 아카이브에서 받을 수 있는 파일 중, sample-02.txt는 서브태스크 1의 조건을, sample-03.txt는 서브태스크 4의 조건을 만족한다.