**문제**

# A번 - 부당한 퍼즐 성공

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **시간 제한** | **메모리 제한** | **제출** | **정답** | **맞은 사람** | **정답 비율** |
| 2 초 | 256 MB |  |  |  |  |

## 문제

현욱은 퍼즐 게임을 굉장히 좋아한다. 어느날 현욱은 친구로부터 간단한 플래시 퍼즐 게임을 하나 추천 받았는데, 이 퍼즐 게임은 다음과 같은 규칙을 갖고 있다.

1. 플레이어는 1 ~ n 까지 숫자가 한 번씩만 나타나는 수열을 하나 가지고 시작한다.
2. 또 다른 1 ~ n 까지 숫자가 한 번씩만 나타나는 수열이 주어졌을 때, 처음 수열을 적절히 변형해서 처음 받은 수열을 이 수열과 동일한 수열로 만들어야 한다.
3. 이 때, 플레이어가 수열에 대해서 할 수 있는 동작은 다음 두 가지가 있다. 동작은 몇 번이라도 수행할 수 있다.
   * 뒤집기 : 현재 수열을 거꾸로 뒤집는다. ex) 1 2 3 4 5 -> 5 4 3 2 1
   * 밀기 : 현재 수열을 왼쪽 혹은 오른쪽으로 한 칸 민다. ex) 1 2 3 4 5 -> 5 1 2 3 4

퍼즐을 풀던 현욱은 분명히 엄청 쉬운 규칙인데도 불구하고 문제가 안 풀려서, 한참을 고민하다가 다시 잘 비교해보니 정답 수열을 주어진 동작만으로는 절대 만들 수가 없는 문제였다!

화가 난 현욱은 퍼즐 제작자에게 따지기 위해 주어진 문제가 올바른 문제인지 아닌지 확인하는 프로그램을 만들기로 결심했다. 현욱을 도와 괘씸한 퍼즐 제작자를 응징해주자.

## 입력

첫째 줄에 n이 주어진다(1 ≤ n ≤ 1,000,000).

둘째 줄에 1에서 n까지의 수가 한 번만 나타나는 수열이 순서대로 주어진다.

셋째 줄에 주어진 두 연산을 수행해서 구성할 수 있는지 확인할 1에서 n까지 수가 한 번만 나타나는 수열이 순서대로 주어진다.

## 출력

주어진 두 가지 연산만을 가지고 처음 수열을 결과 수열로 만들 수 있다면 good puzzle, 아니면 bad puzzle을 출력한다.

## 예제 입력 복사

5  
1 2 3 4 5  
4 3 2 1 5

## 예제 출력 복사

good puzzle

## 예제 입력 2 복사

5  
1 2 3 4 5  
1 2 4 3 5

## 예제 출력 2 복사

bad puzzle

**ANSWER -1 런타임 에러 size 1 에대한 처리를안해줌**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int check = scan.nextInt();

int[] test = new int[1000000];

int[] origin = new int[check];

int[] change = new int[check];

for (int i = 0; i < check; i++) {

origin[i] = scan.nextInt();

}

for (int i = 0; i < check; i++) {

change[i] = scan.nextInt();

}

for (int i = 0; i < check; i++) {

if (origin[0] == change[i]) {

check = i;

break;

}

}

int vector = 0;

if(origin.length - 1 == check){

if(origin[1] == change[0]){

vector ++;

check++;

}else if(origin[1] == change[check-1]){

vector --;

check--;

}else{

System.out.println("bad puzzle");

System.exit(1);

}

}else if(check ==0){

if(origin[1] == change[1]){

vector ++;

check++;

}else if(origin[1] == change[origin.length - 1]){

vector --;

check--;

}else{

System.out.println("bad puzzle");

System.exit(1);

}

}else{

if(origin[1] == change[check+1]){

vector ++;

check++;

}else if(origin[1] == change[check - 1]){

vector--;

check--;

}else{

System.out.println("bad puzzle");

System.exit(1);

}

}

for(int i=1;i<origin.length;i++){

if(check >= origin.length ) check = 0;

if(check < 0 ) check = origin.length -1;

if(origin[i] != change[check]){

System.out.println("bad puzzle");

System.exit(1);

}

check += vector;

}

System.out.println("good puzzle");

scan.close();

}

}

**ANSWER -2 GOOD**

package algorithm;

import java.util.Scanner;

public class A {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int check = scan.nextInt();

int[] origin = new int[check];

int[] change = new int[check];

for (int i = 0; i < check; i++) {

origin[i] = scan.nextInt();

}

for (int i = 0; i < check; i++) {

change[i] = scan.nextInt();

}

if (origin.length == 1) {

if (origin[0] == change[0]) {

System.out.println("good puzzle");

System.exit(1);

} else {

System.out.println("bad puzzle");

System.exit(1);

}

}

for (int i = 0; i < check; i++) {

if (origin[0] == change[i]) {

check = i;

break;

}

}

int vector = 0;

if (origin.length - 1 == check) {

if (origin[1] == change[0]) {

vector++;

check++;

} else if (origin[1] == change[check - 1]) {

vector--;

check--;

} else {

System.out.println("bad puzzle");

System.exit(1);

}

} else if (check == 0) {

if (origin[1] == change[1]) {

vector++;

check++;

} else if (origin[1] == change[origin.length - 1]) {

vector--;

check--;

} else {

System.out.println("bad puzzle");

System.exit(1);

}

} else {

if (origin[1] == change[check + 1]) {

vector++;

check++;

} else if (origin[1] == change[check - 1]) {

vector--;

check--;

} else {

System.out.println("bad puzzle");

System.exit(1);

}

}

for (int i = 1; i < origin.length; i++) {

if (check >= origin.length)

check = 0;

if (check < 0)

check = origin.length - 1;

if (origin[i] != change[check]) {

System.out.println("bad puzzle");

System.exit(1);

}

check += vector;

}

System.out.println("good puzzle");

scan.close();

}

}