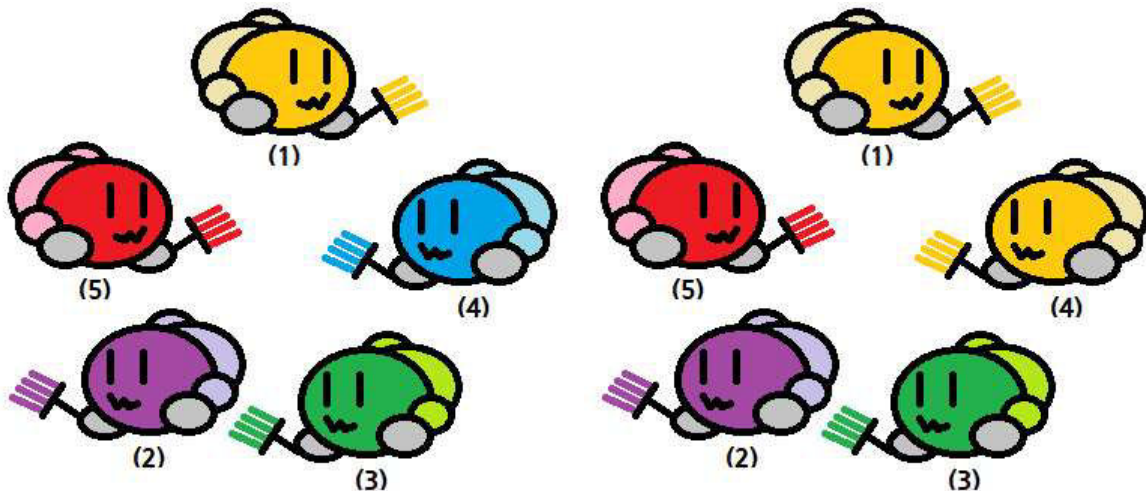


## 8. 1-Color Coloring

문제 유형 : Interactive

시간 제한 : 1 초, 메모리 제한 : 64 MB

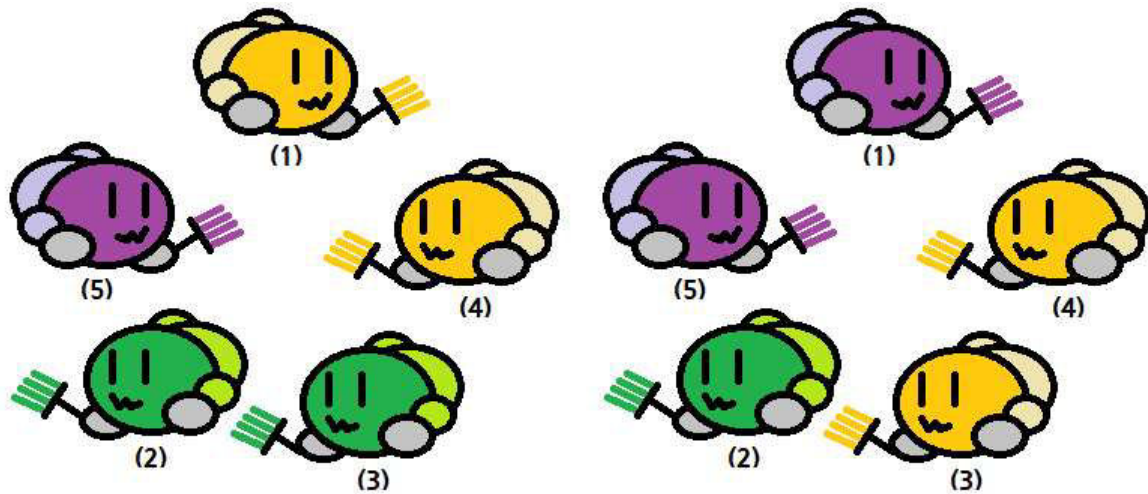
N마리의 정령들이 원형으로 둘러앉아서 색칠놀이를 하려고 한다. 각 정령들은 1, 2, ..., N번의 번호가 붙어있고, i번 정령의 색은 i이다.



각 정령들은 시계방향을 기준으로 다음 위치에 있는 정령을 바라보고 있으며 항상 자신이 바라보는 정령을 색칠한다. A번 색을 가진 정령이 다른 정령을 색칠한다면, 그 정령의 색이 A가 된다. 위 그림에서 1번 정령이 4번 정령을 색칠하면 4번 정령의 색이 노란색(1번 색)이 된다.

유현이는 정령들과 같이 있으며, 재울이는 다른 곳에 있다. 재울이는 메신저로 유현이와 소통하여 정령들을 모두 1번 색으로 칠하려고 한다.

재울이는 유현이를 통해 정령들에게 앞에 있는 정령을 색칠하라는 지시를 내릴 수 있다. 그러면 각 정령들은 자신의 앞에 있는 정령을 자신의 색으로 칠한다. 허나 유현이는 한 번에 최대 한 마리의 정령에게만 지시를 내릴 수 있다.



앞선 상황에서 3번 정령에게 지시를 내리면 왼쪽 그림과 같은 상황이 되며, 그 후 4, 5번 정령에게 지시를 내리면 오른쪽 그림과 같은 상황이 된다.

재울이는 유현이에게 특정 색 정령이 있는지 물어볼 수 있다. 그러면 유현이는 정령이 모여있는 모습을 대강 보고 ‘있다’, ‘없다’ 중 한 가지 대답을 한다. 유현이는 한 번에 여러 색을 셀만큼 머리가 좋지 않기 때문에 오직 한 종류의 색에 대해서만 개수를 셀 수 있다. 정령을 세는데 시간이 꽤 걸리므로 그 사이에 정령들은 자기 본연의 색으로 돌아온다. 위 상황에서 재울이가 노란 정령이 있는지 물어본다면 유현이는 ‘노란 정령이 있다’라고 답을 하며, 정령을 센 후 1, 2, 3, 4, 5번 정령들은 노랑, 보라, 초록, 파랑, 빨강으로 변한다. 유현이가 색깔을 세기 전까지는 정령의 색이 변하지 않으므로 유현이의 대답이 틀릴 가능성은 없다.

유현이가 정령들의 배치 상황을 찍어서 재울이에게 보내면 좋을 테지만, 유현이가 남은 데이터가 얼마 없어서 그럴 수 없다. 즉, 재울이는 정령들의 배치 상황을 모르는 상태에서 모든 정령들을 1번 색으로 바꿔야 한다.

재울이를 도와 모든 정령들을 1번 색으로 바꾸는 프로그램을 짜보자.

## 요구사항

당신은 정령들을 전부 1번 색으로 칠하는 함수 `void ColoringSame(N)` 을 작성해야 한다. (`ColoringSame(N)` 함수는 `coloring.c` / `coloring.cpp`에 있다.)

`void ColoringSame(int N)` 함수의 인수는 다음과 같다.

-  $N$  : 정령의 수 ( $3 \leq N \leq 100$ )

당신은 `ColoringSame` 함수 안에서 `Color(A)`와 `GetColor(C)`를 몇 번 호출할 수 있다.

void Color(int A) 는 A번 정령에게 자기 앞에 있는 정령을 색칠하라고 지시하는 함수이다.

- A : 색칠하라고 지시할 정령의 번호 ( $1 \leq A \leq N$ )

int GetColor(int C) 는 색깔이 C인 정령이 있는지 물어보는 함수이다. 이 함수를 호출한 후 모든 정령의 색이 초기화된다.

- C : 물어볼 색깔의 번호 ( $1 \leq C \leq N$ )

- 반환값 : 색깔이 C인 정령의 수 (없으면 0, 1마리이면 1)

ColoringSame(N) 함수가 끝나면 모든 정령의 색이 1이어야 한다. 만약 그렇지 않다면 채점 프로그램은 'Wrong[1]'을 출력한다. 함수를 호출할 때 인자의 범위를 벗어난다면 'Wrong[2]'를 출력하고 프로그램이 즉시 끝난다. 만약 Color(A)와 GetColor(C)를 특정 횟수를 초과하여 호출하면 'Wrong[3]'을 출력하고 프로그램이 즉시 끝난다.

Color(A) 함수는 최대 7,300번 호출할 수 있으며 GetColor(C) 함수는 최대 200번 호출할 수 있다.

예시 프로그램은 다음과 같다. 주석은 문제와 같은 상황에서 함수의 반환값을 의미한다.

```
#include "coloring.h"

void ColoringSame(int N) {
    Color(1); Color(2); Color(3); Color(4); Color(5);
    int cnt = GetColor(1); // cnt = 1
    Color(1); Color(4); Color(3); Color(2);
}
```

### 샘플 인터페이스

문제 페이지에서 샘플 코드를 다운로드받을 수 있다. 만약 Visual Studio나 Eclipse, Code::Blocks 와 같은 IDE 툴을 사용한다면 coloring.cpp, coloring.h, grader.cpp (또는 coloring.c coloring.h, grader.c)를 한 프로젝트에 넣어서 컴파일하면 된다. 터미널에서 코드를 컴파일한다면 대회 페이지에 있는 컴파일 명령어를 이용하면 된다.

프로그램을 실행한 후 표준입력(stdin)으로 N과 N마리의 정령들의 번호를 입력받으면 샘플 채점기는 정답/오답 여부를 출력한다.

답안을 제출할 때에는 coloring.c 또는 coloring.cpp만 제출하면 된다.

**입출력 예제**

| 입력 (stdin)  | 출력 (stdout) |
|-------------|-------------|
| 5 1 4 3 2 5 | Correct     |

**참고**

메모리 접근, 시스템 호출 등의 비정상적인 방법을 사용하면 오답 처리될 수 있음에 유의하여야.