

하노이의 탑 1

하노이의 탑(Tower of Hanoi) 문제는 잘 알려진 퍼즐 문제로서, 세 개의 막대와 가운데 구멍이 뚫린 여러 개의 크기가 다른 원반(disc)들이 주어질 때 다음과 같은 조건을 만족하면서 원반을 이동시키는 문제이다. (다음 페이지의 그림을 참고할 것)

- 1. 그림 1과 같이, 원반들은 크기가 작은 것이 위에, 크기가 큰 것이 아래에 있도록 정렬되어 (따라서 원뿔 모양과 비슷하게 정렬되어) 1번 막대에 놓여 있음. 크기가 가장 작은 원반이 1번이고, 크기가 클수록 번호가 커짐
- 2. 1회의 '이동' 동작으로 하나의 원반만 현재 막대에서 다른 막대로 이동
- 3. 이동이 가능하기 위해서는, 해당 원반은 현재 위치한 막대에서 가장 위에 있어야 하며, 다른 막대로 이동할 때 더 작은 원반 위에 놓일 수는 없음
- 4. 위와 같은 '이동' 동작을 반복하여 그림 4와 같이 모든 원반을 3번 막대에 크기 순으로 정렬된 상태로 옮기는 것이 목표임
- 5. '이동'의 회수는 최소한으로 하고자 함

예를 들어, 원반의 수가 3개일 때 최적 이동 경로는 다음과 같다.

누적 이동 횟수	동작
1	원반 1: 막대 1 → 막대 3
2	원반 2: 막대 1 → 막대 2
3	원반 1: 막대 3 → 막대 2
4	원반 3: 막대 1 → 막대 3
5	원반 1: 막대 2 → 막대 1
6	원반 2: 막대 2 → 막대 3
7	원반 1: 막대 1 → 막대 3

위의 예에서, 누적 이동 횟수가 4일 때, 막대 2에는 원반 1과 원반 2가 쌓여 있는 상태가 된다.

원반의 개수 N 과 누적 이동 횟수 K 가 주어질 때, 누적 이동 K 회 직후의 막대 2의 상태를 출력하는 프로그램을 작성하시오. 원반을 크기 순으로 1번부터 N 번이라 할 때, 막대 2의 상태는 해당 막대에 놓인 원반의 번호를 맨 위의 것부터 출력하면 된다.

※ 프로그램의 실행 시간은 1 초, 메모리 사용량은 512MB 를 초과할 수 없다.

사용할 수 있는 언어는 C, C++로 제한한다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);
```

단, 위의 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력을 혼용해서 사용하면 안된다. C++의 std::endl 의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n";을 사용하는 것을 권장한다.

Figure 1: 처음 상태

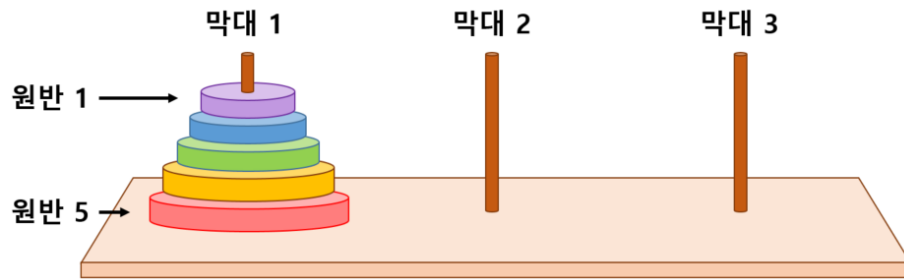


Figure 2: 원반 1을 막대 3으로 옮긴 상태

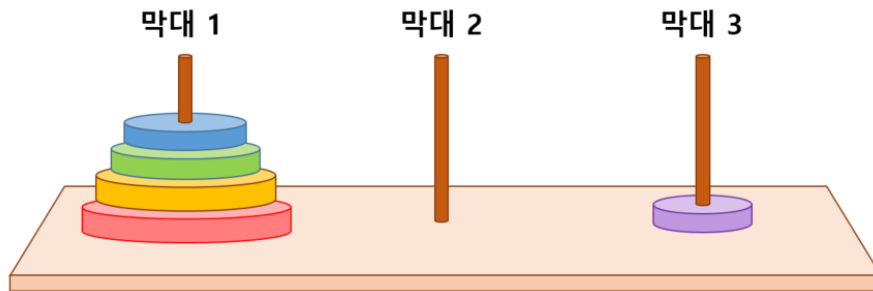
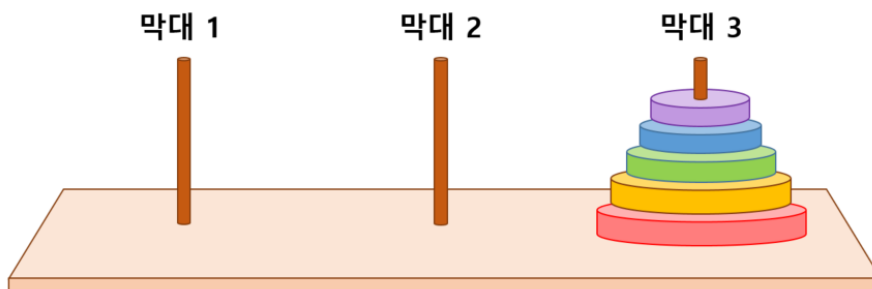


Figure 3: 여러 단계를 거쳐 원반 1~4를 막대 2로 옮긴 상태



Figure 4: 최종 상태



입력

첫 번째 줄에는 테스트 케이스 수 T ($1 \leq T \leq 100$)가 주어진다.

이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 원반의 수 N ($1 \leq N \leq 20$) 과 누적 이동 횟수 K 가 공백으로 구분되어 주어진다. 단, K 는 0보다 크고 N 개의 원반을 막대 1에서 막대 3으로 모두 이동하는 최소 이동 횟수보다는 작거나 같다.

출력

각 테스트 케이스마다, K 번째 이동 동작 직후 막대 2에 있는 원반들의 번호를 맨 위의 것부터 빈 칸을 사이에 두고 차례로 출력한다. 단, 막대 2가 비어 있는 경우에는 0을 출력한다.

예제 입출력

예제 입력	예제 출력
3	1 2
3 4	0
1 1	4
5 23	