

# 마법의 탑

마법의 탑에는 1층부터  $n$ 층까지 각 층마다 괴물이 한 마리씩 봉인되어 있으며, 모든 괴물들은 자연수로 표현되는 능력치를 갖는다. 어느 날, 봉인의 힘이 약해져서 괴물들은 자유롭게 움직일 수 있게 되었지만, 아직 탑을 탈출하기에는 힘이 부족하다.

괴물들은 다른 괴물을 잡아먹어 능력치를 흡수할 수 있으며, 다른 괴물을 잡아먹는 기준은 다음과 같다.

- 괴물은 다른 괴물 한 마리만 잡아먹을 수 있다.
- 자신보다 높은 층에 봉인되었거나 능력치가 높은 괴물은 잡아먹을 수 없다. 단, 능력치의 비교는 최초 능력치만을 기준으로 한다. 예를 들어, 최초 능력치가 괴물 A보다 낮은 괴물 B가 다른 괴물 C를 잡아먹어서 능력치가 높아졌더라도, 괴물 A는 괴물 B를 잡아먹을 수 있다.
- 최대한 많은 능력치를 흡수하기 위해, 자신이 잡아먹을 수 있는 괴물들 중 능력치가 가장 높은 괴물을 잡아먹는다. 그런 괴물이 둘 이상이라면 자신이 봉인된 층수와 가장 가까운 곳에 봉인된 괴물을 잡아먹는다.
- 위 조건에 해당하는 괴물이 존재하지 않는다면, 어떠한 괴물도 잡아먹을 수 없다.

모든 괴물들이 각 층마다 봉인된 괴물의 능력치를 알고 있을 때, 각각 어느 층의 괴물을 잡아먹으려 할지 알아보자. 단, 서로 다른 괴물이 같은 괴물을 잡아먹으려 할 수도 있다.

예를 들어, 총 5개의 층으로 이루어진 탑에 봉인된 괴물들의 능력치가 1층부터 5층까지 차례대로 20, 40, 10, 20, 30일 때, 1, 3층에 봉인된 괴물이 잡아먹을 수 있는 괴물은 존재하지 않는다. 2층의 괴물은 1층의 괴물을 잡아먹으려 할 것이다. 4층의 괴물은 잡아먹을 수 있는 1, 3층의 괴물 중 능력치가 가장 높은 1층의 괴물을 잡아먹으려 할 것이다. 5층의 괴물은 1, 3, 4층의 괴물을 잡아먹을 수 있으며, 이 중 능력치가 가장 높은 괴물 중 자신과 가장 가까운 곳에 봉인된 4층의 괴물을 잡아먹으려 한다.

※ 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.

# 입력

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

이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에 탑의 총 층수  $n$  ( $1 \leq n \leq 100,000$ )이 주어진다.
- 두 번째 줄에 1층부터  $n$  층까지 차례대로  $i$  ( $1 \leq i \leq n$ ) 층 괴물들의 능력치인 자연수  $P_i$  ( $1 \leq P_i \leq 1,000,000$ )가 공백으로 구분되어 주어진다.

# 출력

테스트 케이스마다 1층부터  $n$  층까지 각 괴물들이 잡아먹으려 하는 괴물의 층수를 차례로 출력한다. 단, 잡아먹을 수 있는 괴물이 존재하지 않으면, -1을 출력한다.

## 예제 입출력

예제 입력	예제 출력
2	-1 1 -1 1 4
5	-1 1 2 2 4
20 40 10 20 30	
5	
1 1 1000 1 1	