

## 1. 문제 요약

1. 탑이 여러개 있다.
2. 모든 탑은 왼쪽으로 레이저를 발사한다.
3. 가장 먼저 만나는 단 하나의 탑에서 수신한다.
4. 각각의 탑에서 발사한 레이저 신호를 어느 탑에서 수신하는지 알아내는 프로그램을 작성한다.

## 2. 문제 입/출력

### 입력

1. 첫째 줄에 탑의 수를 나타내는  $N$
2. 둘째 줄에  $N$ 개의 탑들의 높이가 하나의 공백을 두고 주어진다.

### 출력

1. 주어진 탑들의 순서대로 각각의 탑들에서 발사한 레이저 신호를 수신한 탑들의 번호를 하나의 공백을 두고 출력

## 3. 입력 제한 크기

1. 정수  $N \leq 500,000$
2. 정수  $H \leq 100,000,000$

## 4. 접근 아이디어

1. 탑에서 발사한 레이저가 왼쪽의 어느 빌딩에 닿는지 확인해야하므로, 해당 빌딩을 기록해야한다.
2. 조건에 따라 레이저가 닿는 빌딩을 찾아야 하므로, 순서가 있으며 조건에 해당하지 않는 빌딩 정보를 지울 수 있는 Stack 사용

## 5. 시간 복잡도

$O(N)$  최대 500,000번 입력

## 6. 구현 코드

```
package org.example;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.*;

public class Main {
    private static StringBuilder sb = new StringBuilder();
    private static int[] arr;
    private static int[] results;

    // Index와 Value를 같이 저장할 클래스 정의
```

```
static class IndexAndValue {
    int index;
    int value;

    public IndexAndValue(int index, int value) {
        this.index = index;
        this.value = value;
    }
}

public static void main(String[] args) throws IOException {
    System.setIn(new FileInputStream("input.txt"));
    BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

    // N 입력
    int N = Integer.parseInt(br.readLine());

    // 배열 초기화
    arr = new int[N + 1];
    results = new int[N + 1];

    // 빌딩 정보 입력
    String[] split = br.readLine().split(" ");
    for (int i = 1; i <= N; i++) {
        arr[i] = Integer.parseInt(split[i - 1]);
    }

    // stack 선언
    Stack<IndexAndValue> stack = new Stack<>();

    // index 1부터 시작
    a: for (int i = 1; i <= N; i++) {
        // top보다 큰 값이 들어오면
        if (stack.size() > 0) {
            // stack에 값이 없을 때까지, arr[i]보다 작은 값들 다 빼줌
            while (stack.peek().value < arr[i]) {
                stack.pop();

                if (stack.isEmpty()) {
                    stack.push(new IndexAndValue(i, arr[i]));
                    continue a;
                }
            }

            IndexAndValue peek = stack.peek();
            results[i] = peek.index;
        }

        // stack에 빌딩을 넣어준다.
        stack.push(new IndexAndValue(i, arr[i]));
    }

    // StringBuilder에 결과를 넣어준다.
```

```
        for (int i = 1; i <= N; i++) {
            sb.append(results[i]).append(" ");
        }

        // 출력
        System.out.println(sb);
    }
}
```