

Day 87 coding Statement :

There are N stones in a pond, each having a value A_i written on it. A frog is at stone 1 and wants to reach stone N . The frog can jump from a stone i to any stone j ($j > i$). Let d be the length of subarray (i.e. $j - i + 1$), then the energy required for the jump is $(d \cdot A_i) - A_j$. Find the minimum **non-negative** amount of energy required by the frog to reach the N -th stone.

Note: It is possible that the **total** amount of energy required is negative, in that case, you should print the minimum non-negative value (i.e. 0).

Input Format

- The first line contains an integer T - the number of test cases. Then the test cases follow.
- The first line of each test case contains an integer N - the number of stones.
- The second line contains N integers denoting the numbers written on the stones.

Output Format

For each test case output a single integer - the minimum **non-negative** energy required by the frog.

Sample Input

4

3

6 1 3

4

3 1 10 4

3

7 9 1

2

1 5

Sample Output

10

4

20

0

```
import java.util.*;
import java.lang.*;
import java.io.*;

class Main {
    public static void main(String[] args) throws java.lang.Exception {
        BufferedReader bu = new BufferedReader(new
InputStreamReader(System.in));
        StringBuilder sb = new StringBuilder();
        int t = Integer.parseInt(bu.readLine());
        while (t-- > 0) {
            int n = Integer.parseInt(bu.readLine());
            String s[] = bu.readLine().split(" ");
            int a[] = new int[n], i;
            for (i = 0; i < n; i++)
                a[i] = Integer.parseInt(s[i]);
            long ans = a[0];
            int min = a[0];
            for (i = 1; i < n; i++) {
                ans += min;
                min = Math.min(min, a[i]);
            }
            ans -= a[n - 1];
            ans = Math.max(ans, 0);
            sb.append(ans + "\n");
        }
        System.out.print(sb);
    }
}
```