Maximum Difference (`maxdifference`)

You are given an array A of N integers. Your goal is to split the array into one or more non-empty contiguous subarrays.

The **value** of a subarray is defined as the difference between its maximum and minimum elements.

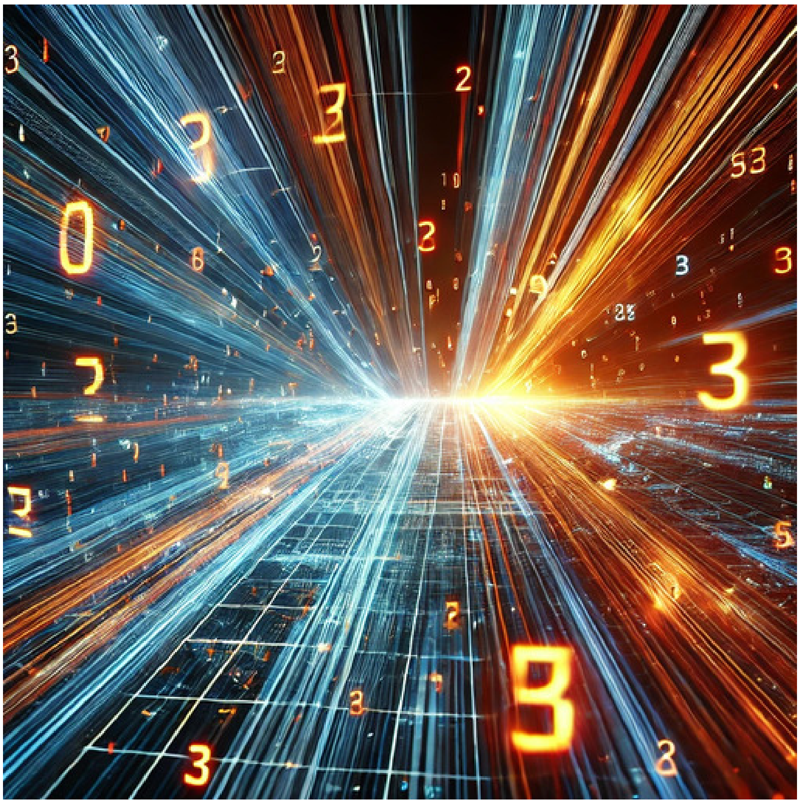

Figure 1: Can you find an optimal split?

Your task is to determine the best way to split the array to maximize the sum of these values. Write a program which computes these optimal sums!

> ☞ Among the attachments of this task you may find a template file `maxdifference.*` with a sample incomplete implementation.

Input

The first line of the input file contains a single integer T, the number of test cases. T test cases follow.

Each test case consists of:

- a line containing a single integer N.
- a line containing the array A: $A_0, A_1, \ldots A_{N-1}$.

Output

The output file must contain T lines corresponding to the test cases, each consisting of integer P, the maximum sum of subarrays with an optimal construction.

Constraints

- $1 \leq N \leq 200\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000$ for each $i = 0 \ldots N-1$.
- The sum of N across all testcases does not exceed $200\,000$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points)	Examples.
– **Subtask 2** (30 points)	The sum of N across all testcases does not exceed 5000.
– **Subtask 3** (30 points)	$1 \leq A_i \leq 2$.
– **Subtask 4** (40 points)	No additional limitations.

Examples

input	output
7 4 2 1 4 3 5 1 2 2 1 2 6 1 3 6 2 4 5 6 1 4 6 2 5 3 10 7 1 10 9 4 2 8 5 3 6 10 3 1 4 1 5 9 2 6 5 3 6 1000000000 1 1000000000 1 1000000000 1	3 2 8 9 23 17 2999999997

Explanation

In the **first testcase of the example**, the value of the whole array is $4 - 1 = 3$.

In the **second testcase**, splitting A into $[1, 2]$ and $[2, 1, 2]$ gives the total value $1 + 1 = 2$.

In the **third testcase**, splitting A into $[1, 3, 6]$, and $[2, 4, 5]$ gives the total value $3 + 5 = 8$.

It can be proven that the total values above are optimal.