Practice Mode

Contest scoreboard | Sign in

Round 1A 2010

A. Rotate

B. Make it Smooth

C. Number Game

Contest Analysis

Questions asked 1

- Submissions

Rotate

11pt	Not attempted **2076/2436 users** correct (85%)
12pt	Not attempted **1855/2071 users** correct (90%)

Make it Smooth

12pt	Not attempted **509/954 users** correct (53%)
24pt	Not attempted **319/482 users** correct (66%)

Number Game

16pt	Not attempted **680/1091 users** correct (62%)
25pt	Not attempted **244/450 users** correct (54%)

- Top Scores

rng..58	100
Pipi	100
cgy4ever	100
rem	100
XiaoZiqian	100
qizichao	100
exod40	100
GarnetCrow	100
hos.lyric	100
ACRush	100

Problem B. Make it Smooth

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 12 points	Solve B-small
Large input 24 points	Solve B-large

Problem

You have a one-dimensional array of **N** pixels. Each pixel has a value, represented by a number between 0 and 255, inclusive. The *distance* between two pixels is the absolute difference of their numbers.

You can perform each of the following operations zero or more times:

1. With cost **D**, delete any pixel, so its original neighbors become neighboring pixels.
2. With cost **I**, insert one pixel of any value into any position -- either between two existing pixels, or before the first pixel, or after the last pixel.
3. You can change the value of any pixel. The cost is the absolute difference of the old value of the pixel and the new value of the pixel.

The array is *smooth* if any neighboring pixels have distance at most **M**. Find the minimum possible cost of a sequence of operations that makes the array smooth.

Note: The empty array -- the array containing no pixels -- is considered to be smooth.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow, each with two lines. The first line is in the form "**D I M N**", the next line contains **N** numbers $\mathbf{a_i}$: the values of the pixels from left to the right.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1), and y is the minimum cost to make the input array smooth.

Limits

All the numbers in the input are integers.
$1 \le \mathbf{T} \le 100$
$0 \le \mathbf{D}, \mathbf{I}, \mathbf{M}, \mathbf{a_i} \le 255$

Small dataset

$1 \le \mathbf{N} \le 3$.

Large dataset

$1 \le \mathbf{N} \le 100$.

Sample

```
Input         Output

2             Case #1: 4
6 6 2 3       Case #2: 17
1 7 5
100 1 5 3
1 50 7
```

Explanation

In Case #1, decreasing the 7 to 3 costs 4 and is the cheapest solution. In Case #2, deleting is extremely expensive; it's cheaper to insert elements so your final array looks like `[1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 50, 45, 40, 35, 30, 25, 20, 15, 10, 7]`.