

Candies

<https://codingcompetitions.withgoogle.com/kickstart/round/000000000019ff43/0000000000337b4d>

Problem

Carl has an array of N candies. The i -th element of the array (indexed starting from 1) is A_i representing sweetness value of the i -th candy. He would like to perform a series of Q operations. There are two types of operation:

- Update the sweetness value of a candy in the array.
- Query the sweetness score of a subarray.

The sweetness score of a subarray from index l to r is: $A_l \cdot 1 - A_{l+1} \cdot 2 + A_{l+2} \cdot 3 - A_{l+3} \cdot 4 + A_{l+4} \cdot 5 \dots$. More formally, the sweetness score is the sum of $(-1)^{i-l} \cdot A_i \cdot (i-l+1)$, for all i from l to r inclusive. For example, the sweetness score of:

- $[3, 1, 6]$ is $3 \cdot 1 - 1 \cdot 2 + 6 \cdot 3 = 19$
- $[40, 30, 20, 10]$ is $40 \cdot 1 - 30 \cdot 2 + 20 \cdot 3 - 10 \cdot 4 = 0$
- $[2, 100]$ is $2 \cdot 1 - 100 \cdot 2 = -198$

Carl is interested in finding out the total sum of sweetness scores of all queries. If there is no query operation, the sum is considered to be 0. Can you help Carl find the sum?

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case begins with a line containing N and Q . The second line contains N integers describing the array. The i -th integer is A_i . The j -th of the following Q lines describe the j -th operation. Each line begins with a single character describing the type of operation (**U** for update, **Q** for query).

- For an update operation, two integers X_j and V_j follow, indicating that the X_j -th element of the array is changed to V_j .
- For a query operation, two integers L_j and R_j follow, querying the sweetness score of the subarray from the L_j -th element to the R_j -th element (inclusive).

Output

For each test case, output one line containing **Case #x: y**, where **x** is the test case number (starting from 1) and **y** is the total sum of sweetness scores of all the queries.

Limits

- Time limit: 20 s per test set.
- Memory limit: 1 GB.
- $1 \leq T \leq 100$.
- $1 \leq A_i \leq 100$, for all i .
- $1 \leq N \leq 2 \cdot 10^5$ and $1 \leq Q \leq 10^5$ for at most 6 test cases.

- For the remaining cases, $1 \leq N \leq 300$ and $1 \leq Q \leq 300$.
- If the j -th operation is an update operation, $1 \leq X_j \leq N$ and $1 \leq V_j \leq 100$.
- If the j -th operation is a query operation, $1 \leq L_j \leq R_j \leq N$.

Test set 1

There will be at most 5 update operations.

Test set 2

There are no special constraints.

Sample

Input

```
2
5 4
1 3 9 8 2
Q 2 4
Q 5 5
U 2 10
Q 1 2
3 3
4 5 5
U 1 2
U 1 7
Q 1 2
```

Output

```
Case 1: -8
Case 2: -3
```

Minové pole

<https://kasiopea.matfyz.cz/archiv/2016/finale/F>

Po jedné bitvě zůstali na bojišti miny a čeští (protitankoví) jezkové. My chceme oplotit všechny miny ostnatým drátem aby nám nikdo nevybouchl. Plot chceme připevnit přímo na ježky, takže si ho můžeme představit jako mnohoúhelník s vrcholy v ježcích ve kterém leží všechny miny. Pozor, miny musí ležet opravdu uvnitř, ne na samotném plotě. Pomozte nám zjistit, kolik nejméně pletiva potřebujeme.

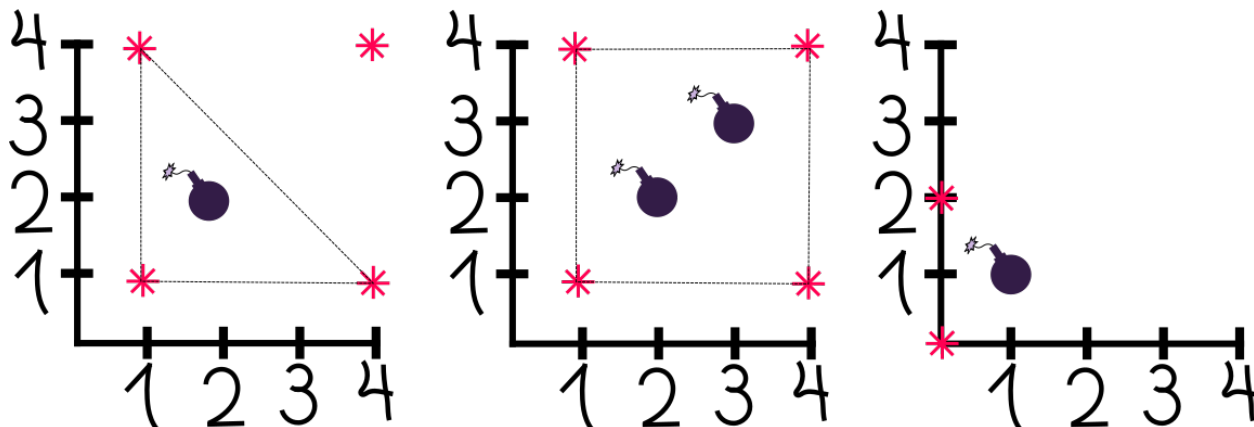


Figure 1: Znázornění vzorových vstupů

Vstup

Na prvním řádku se nachází číslo T , počet testů. Na dalších řádcích je postupně T testů, každý v následujícím tvaru: První řádek testu obsahuje celé kladné číslo H , počet ježků. Druhý obsahuje M , celé kladné číslo reprezentující počet min. Na dalších H řádcích jsou souřadnice ježků. Vždy na i -tém řádku jsou dvě celá čísla $x_i y_i$ popisující souřadnice i -tého ježka. Následujících M řádků obsahuje souřadnice min ve stejném formátu. Můžete předpokládat, že všechny souřadnice jsou v rozmezí -10^8 až 10^8 .

Výstup

Když se miny nedají oplotit vypište na jediný řádek číslo -1 . Jinak vypište jedno reálné číslo a to obvod nejmenšího možného oplocení. Maximální povolená chyba je 10^{-7} .

Lehká verze

- $T \leq 10$
- $H \leq 20$
- $M \leq 1$

Těžká verze

- $T \leq 10$
- $H \leq 300$
- $M \leq 1000$

Příklad

Vstup

3
4
1
1 1
4 1
4 4
1 4
2 2
4
2
1 1
4 1
4 4
1 4
2 2
3 3
2
1
0 0
0 2
1 1

Výstup

10.24264068711928
12
-1