

Another Update-Query Problem

Limits: 1s, 512 MB

You will be given an array **A** of length **N**. On that array you will have to do **Q** operations. Operations are of two types.

Operation-1: U(l, r, x): Add **x** to **A_i**, where **i** is in range **[l, r]**

Operation-2: Q(l, r, d): Print the output of this series modulo **1000000007**:

$$1A_l + (1+d)A_{l+1} + (1+2d)A_{l+2} + (1+3d)A_{l+3} + \dots + (1+(r-l)d)A_r$$

Input

Input starts with an integer **T** ($1 \leq T \leq 10$), denoting the number of test cases.

First line of each case has two integers, **N** ($1 \leq N \leq 100000$) and **Q** ($1 \leq Q \leq 100000$).

Second line has **N** integers **A_i** ($1 \leq A_i \leq 1000000000$), indicating the values of the array.

Following **Q** lines each has four integers. First of those four integers are **C** ($1 \leq C \leq 2$).

If **C=1**, then it requests an operation of type-1. The other three integers will be **L** ($1 \leq L \leq N$), **R** ($1 \leq R \leq N$, $L \leq R$), **X** ($1 \leq X \leq 1000000000$). And you will have to do operation **U(L, R, X)**.

If **C=2**, then it requests an operation of type-2. The other three integers will be **L** ($1 \leq L \leq N$), **R** ($1 \leq R \leq N$, $L \leq R$), **D** ($1 \leq D \leq 1000000000$). And you will have to print the value of **Q(L, R, D)** modulo **1000000007**.

Easy Subtask:

$$1 \leq N \leq 2000$$

$$1 \leq Q \leq 2000$$

C=2 for all cases. That means there is no request for operation-1.

Medium Subtask:

$$1 \leq N \leq 100000$$

$$1 \leq Q \leq 100000$$

C=2 for all cases. That means there is no request for operation-1.

Hard Subtask:

Full specification

Output

For each case, first print the case number, starting from 1, in a separate line. For each request of operation-2, print the result in a separate line.

Samples

Input

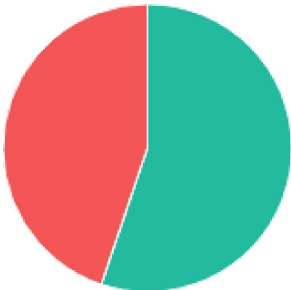
Output

Input	Output
2	
5 4	
2 8 19 9 1	Case 1:
2 2 5 3	157
2 2 4 6	258
2 2 5 3	157
2 2 5 8	357
3 3	Case 2:
6 7 1	7
2 2 2 7	119
1 1 3 8	
2 1 2 6	

Notes

- 1. Sample case-2 is not applicable for easy and medium subtasks.
- 2. Data set is huge, use fast I/O methods.

Statistics



Earliest (/s/46321), 15w ago	lamHot (/u/lamHot)
Fastest (/s/46368), 0.1s	bakupito (/u/bakupito)
Lightest (/s/46393), 4.7 MB	Rajib_119 (/u/Rajib_119)
Shortest (/s/48054), 2462B	reyadussalahin (/u/reyadussalahin)

Submit