Problem C – Colorful Polygon

The *Legendary Huron* owns a long, narrow box filled with magnetic sticks. Each stick has a color. Colors are represented by integers from 1 to 10^6.

The box can only store sticks in a single line, from left to right. Initially, the box contains n sticks. You are given an array A of length n, where A_i is the color of the i-th stick from left to right.

Over time, the Legendary Huron will perform $Q_1 + Q_2$ operations on his box. There are four possible operations:

1. Buy a bag of k sticks of color c and insert them to the **left end** of the box.
2. Buy a bag of k sticks of color c and insert them to the **right end** of the box.
3. Remove the **leftmost** k sticks from the box and give them to his cousin, Huroncito.
4. Remove the **rightmost** k sticks from the box and give them to Huroncito.

After each operation, the Legendary Huron will take the sticks from his box to play with them and attempt to build a **colorful polygon** using the sticks he currently possesses. A polygon is *colorful* if it satisfies the following conditions:

- It must be simple with no self-intersections and have a positive area.
- Each side of the polygon must consist of sticks of the same color.
- No two sides may share the same color.
- The length of a side equals the number of sticks used for that side.

For example, if the box has the following sticks $[4, 3, 1, 1, 2, 3, 3, 4, 4]$, where different integers represent different colors, a valid colorful polygon could be a triangle. One side may consist of two sticks of color 1, another side of three sticks of color 3, and the last side of two sticks of color 4. The total perimeter in this case is 7. Note that it is not necessary to use all colors or all sticks of a color.

Each time he plays with the sticks, the Legendary Huron asks for the maximum perimeter of a **colorful polygon** that can be constructed. If no colorful polygon can be formed, the result is 0. After playing with the sticks, they are stored in the same order as before in the box.

The first Q_1 operations are given explicitly in the input. The last Q_2 operations, however, are not given directly. Instead, they are generated **using a random generator described in the Input section**. **Read the Output section to see how to print the answer for the last Q_2 operations**.

Input

The first line contains three integers n, Q_1 and Q_2 ($1 \le n, Q_1 \le 3 \cdot 10^5$ and $1 \le Q_2 \le 5 \cdot 10^7$).

The second line contains n integers $A_1, A_2, \ldots, A_n$ ($1 \le A_i \le 10^6$) — the initial colors of the sticks in the box.

The next Q_1 lines describe the first sequence of operations:

- If the operation type is 1 or 2, the line contains three integers t, k, and c ($t \in \{1, 2\}$, $1 \le k \le 10^9$, and $1 \le c \le 10^6$).
- If the operation type is 3 or 4, the line contains two integers t and k ($t \in \{3, 4\}$ and $0 \le k \le \min(10^9, sz)$, where sz is the current amount of sticks in the box at the moment of the operation).

After these lines, the input specifies the random generator for the second sequence of operations. The first line of the random generator specification contains two integers a and d ($0 \le a, d < 10^9 + 7$) — the parameters of the generator described below.

The next line contains X_0 ($0 \le X_0 < 10^9 + 7$) — the initial seed.

The next line contains an integer p ($1 \le p \le 10^6$) — the size of an array used to generate the last Q_2 operations as described below.

The last line in the input contains p integers $B_1, B_2, \ldots B_p$ ($1 \le B_i \le 10^6$) — an array used to generate the last Q_2 operations as described below.

The random values used to generate the last Q_2 operations are produced by the following linear congruential generator (LCG):

$$X_{i+1} = (a \cdot X_i + d)\%(10^9 + 7)$$

The generator is initialized with X_0. Each call to $next()$ returns the next value X_i, starting with X_0.

Each of the last Q_2 operations is then generated as follows:

1. The type is $(next()\%4) + 1$.
2. If the type is 1 or 2:
   - $k = next() + 1$.
   - $c = B[next()\%p) + 1]$.
3. If the type is 3 or 4: $k = next()\%(sz + 1)$, where sz is the current amount of sticks in the box at the moment of the operation.

You can see an example of the generation of the last Q_2 operations in the notes.

Output

Print two lines.

In the first line, print Q_1 space-separated integers — the answer for the first Q_1 operations.

In the second line, print a single integer:

$$\bigoplus_{i=1}^{Q_2} (i \cdot res_i\%998244353)$$

where res_i is the result of the i-th of the last Q_2 operations, and $\oplus$ denotes the bitwise XOR.

Note that only $i \cdot res_i$ is taken modulo 998244353, not the XOR sum.

Note that this modulo is different from the modulo in the generator.

Sample input 1	Sample output 1

<pre>5 4 3
1 1 2 3 3
4 1
3 1
2 2 4
3 2
2 3
7
4
3 1 2 4</pre>
	<pre>3 3 5 0
	15</pre>

Sample input 2	Sample output 2

`10 1 100`
`1 2 3 4 5 6 7 8 9 10`
`2 1 11`
`999983 100003`
`57357831`
`11`
`1 2 3 4 5 6 7 8 9 10 11`
	`11`
	`1023060806`

Sample input 3	Sample output 3

`5 1 1`
`1 2 3 2 1`
`1 1 3`
`0 0`
`1`
`2`
`1 2`
	`6`
	`7`

Note

Explanation of the first sample

After the first $Q_1 = 4$ operations of the example 1, the colors of the sticks in the box are as follows from left to right: $[3, 4, 4]$.

The parameters of the generator are $a = 2$ and $d = 3$. The initial seed is $X_0 = 7$. The next values of X_i are calculated by the expression $X_{i+1} = (2X_i + 3)\%(10^9 + 7)$. The values of $X_0, X_1, \ldots, X_7$ are $7, 17, 37, 77, 157, 317, 637, 1277$, respectively. The last $Q_2 = 3$ operations are generated as follows:

1. The first of the last Q_2 operations is of type $t = X_0\%4+1 = 4$, with $k = X_1\%(3+1) = 1$. The answer after this operation is 0.

2. The following operation is of type $t = X_2\%4 + 1 = 2$, with $k = X_3 + 1 = 78$ and $c = B[X_4\%4 + 1] = 1$. The answer after this operation is 3.

3. The last operation is of type $t = X_5\%4 + 1 = 2$, with $k = X_6 + 1 = 638$ and $c = B[X_7\%4 + 1] = 1$. The answer after this operation is 3.

The last integer in the output, then, should be $(1 \cdot 0) \oplus (2 \cdot 3) \oplus (3 \cdot 3) = 15$.