The 2019 Aisa Nanchang First Round Online Programming Contest

Statements

September, 8, 2019

Problem A. Enju With math problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Enju is a vibrant girl, and she is fond of math problem.

Today, she gives you an array $a_1, a_2, \ldots a_{100}$. Now she is wondering if there exist an integer pos such that the following equation is true:

$$\forall x \in [1, 100], a_x = \varphi(x + pos - 1)$$

$\varphi(x)$ is the number of integers in $[1, x]$ which is coprime with x.

Input

The first line of the input contains an integer T $(1 \leq T \leq 500)$ − the number of the test cases.

For each test cases, it contains 10 lines, 10 positive integers per line, representing the array a.

It is guaranteed that any number in array a is no lesser than 1 and no greater than $1.5 * 10^8$.

Output

For each test cases, if such an integer pos exists, you should output two lines. You should output 'YES' (without quota) in the first line. And you should output pos in the second line. It is guaranteed that if such a pos exists, its' value is no lesser than 1 and no greater than $1.5 * 10^8$.

If such an integer pos doesn't exist, you should output 'NO' (without quota) in one line.

Examples

standard input	standard output
1 1 1 2 2 4 2 6 4 6 4 10 4 12 6 8 8 16 6 18 8 12 10 22 8 20 12 18 12 28 8 30 16 20 16 24 12 36 18 24 16 40 12 42 20 24 22 46 16 42 20 32 24 52 18 40 24 36 28 58 16 60 30 36 32 48 20 66 32 44 24 70 24 72 36 40 36 60 24 78 32 54 40 82 24 64 42 56 40 88 24 72 44 60 46 72 32 96 42 60 40	YES 1
1 1 1 2 2 4 4 6 4 6 4 10 4 12 6 8 8 16 6 18 8 12 10 22 8 20 12 18 12 28 8 30 16 20 16 24 12 36 18 24 16 40 12 42 20 24 22 46 16 42 20 32 24 52 18 40 24 36 28 58 16 60 30 36 32 48 20 66 32 44 24 70 24 72 36 40 36 60 24 78 32 54 40 82 24 64 42 56 40 88 24 72 44 60 46 72 32 96 42 60 40	NO

Problem B. Fire-Fighting Hero

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	128 megabytes

This is an era of team success, but also an era of heroes. Throughout the ages, there have been numerous examples of using the few to defeat the many. There are V (Numbers 1 to V) fire-fighting points in ACM city. These fire-fighting points have E roads to communicate with each other. Among them, there is a fire-fighting hero in the S fire-fighting point, and the fire-fighting team is distributed in K fire-fighting points. If a fire-fighting point needs to be put out, the fire-fighting hero or the fire-fighting team must arrive as soon as possible, that is, to choose the shortest route to arrive.

Today, our fire-fighting heroes want to challenge the fire-fighting team. The challenge is to: The maximum value of the shortest path for a fire-fighting hero to go to others fire-fighting points is compared with the maximum value of the shortest path for a fire-fighting team to go to others fire-fighting points from any point in their fire-fighting points. Because firefighting heroes are different and run faster, the maximum value of the shortest path they get should be discounted first, that is, multiplied by a coefficient of $\frac{1}{C}$, and then compared. The smaller one wins. Who is the real firefighter in this situation?

Input

The first line contains a positive integer $T(1 \leq T \leq 10)$, which indicates that there are T cases of test data.

The format of each case of test data is as follows:

- Line 1 contains five positive integers $V(1 \leq V \leq 1000)$, $E(V-1 \leq E \leq \frac{V*V}{2})$, $S(1 \leq S \leq V)$, $K(1 \leq K \leq V)$ and $C(1 \leq C \leq 10)$, The meaning is shown above.
- Line 2 contains K positive integers, which in turn denotes the location number of the fire-fighting point where the fire-fighting team is located.

In the next E line, three positive integers $i, j(1 \leq i, j \leq V)$ and $L(1 \leq L \leq 10000)$ per line. Represents a path, i, j as the endpoint (fire-fighting point), L as the length of the path.

Output

Each case of test data outputs one line, which is a positive integer. That is, the maximum value of the shortest path of the winner (If the fire hero wins, the maximum value before the discount should be output). A draw is also a victory for fire-fighting hero.

Example

standard input	standard output
1 4 7 3 2 2 1 4 1 2 7 1 3 2 1 4 6 2 1 1 2 4 1 3 2 1 3 4 3	2

Problem C. Hello 2019

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

A digital string is "good": when it contains a subsequence 9102 and does not contain a subsequence 8102.

The bad value of a string is defined as how many characters are to remove at least, so that the string satisfies the "good" property. Output -1 if the string cannot satisfy the "good" property by removing some characters (0 or maybe more).

Input

The first line contains two integers n, Q ($1 \le n, Q \le 2 * 10^5$). Where n is the length of the string and Q is the number of queries.

The second line contains a string s that consists entirely of decimal numbers.

The next Q line, each line contains two integers l, r ($1 \le l \le r \le n$), denoting a query.

Output

For each query, output an answer which is the **bad value** of the substring $s_l\, s_{l+1} \cdots s_r$ from s.

Example

standard input	standard output
8 3 88988102 1 8 2 8 1 7	4 3 -1

Problem D. Interesting Series

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

We define the series F as:

$$F_1 = 1$$
$$F_n = a * F_{n-1} + 1 (n \geq 2)$$

For a set s, we define its value to be $value(s) = F_{sum(s)}$, where $sum(s)$ is the sum of all the elements of the set.

Given a set S having N elements, please calculate the sum of the values of all subsets of S of size K.

$$Answer(K) = \sum_{s \in \ subsets \ of \ S \ and \ |s|=K} value(s)$$

Input

The first line contains three integers N, a, Q $(1 \leq N, Q \leq 10^5, 2 \leq a \leq 1000)$, Q means the Number of queries.

The second line contains N integers,$s_1, s_2, \cdots, s_N$ $(1 \leq s_i \leq 10^9)$.

For the next Q lines, each line contains an integer K,$1 \leq K \leq N$, which means you should output the $Answer(K)$.

Output

For each query, you should output the answer mod (100003).

Example

standard input	standard output
3 2 1 3 1 1 2	33

Note

We have three subsets of size $2 : (3, 1), (3, 1), (1, 1)$. Their values are $15, 15, 3$, and their sum is 33.

Problem E. Magic Master

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

John is not only a magic master but also a shuffling master.

Famous though he is, he likes interacting with his fans by playing a game with his fantastic shuffling skills.

The game shows as follows:

He first shows a deck of pokers contains N cards indexed $1, 2, \ldots, N$ and all the cards are the same. He notes that the side contains numbers as the front side and the other side as the back. Later he selects one of his fans to choose a positive integer M that is not greater than 10. After that, he will shuffle the cards purposefully and put it on the desktop. Note that at this moment the front sides of the cards are facing to the desk.

Then, the selected fans should perform the following steps until there is no card on the desk.

1. Place the card on the top of the piles to John's hand. If there are already some cards in his hand, put it on the top of them.
2. If there remain cards on the desk:
   (a) Put a card from the top of the piles to the bottom of it.
   (b) Execute the step (a) of M times.
   (c) Go to step 1.

Next, it is time to witness the miracle.

John will continually overturn the cards on his hand top to bottom, and we will find that the cards are always in decreasing order.

One day, one of John's fans, Tom, a smart JBer, understands the key to this magic. He turns to you and comments that the key to that magic is the shuffling John did previously. For any number M, he will turn the cards in a specific order after the shuffling. As you are not as smart as Tom, you should find the index of the Kth cards from the top of the piles after the shuffling.

Input

The first line contain a positive integer T $(1 \leq T \leq 10)$ – the number of test cases.

For each test cases, the first line contain a positive integer N $(1 \leq N \leq 40000000)$, indicating the number of cards.

The second line contain a positive integer M $(1 \leq M \leq 10)$ – the number the fans selects.

The third line contain an integer Q $(1 \leq Q \leq 100)$ – indicating that there are Q questions.

Next, there are Q lines, each with an integer K $(1 \leq K \leq N)$ – representing a query.

Output

For each query, you should output the index of the Kth cards from the top of the piles after the shuffling.

Example

standard input	standard output
1 5 1 2 2 3	5 2

Note

Sample description - The indexs of the cards after the shuffling are: 1 5 2 4 3 (Left is top)

Problem F. Megumi With String

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Megumi has a string S of length l. As an inconspicuous girl, she likes playing with string.

She thinks that the **value** of a string str is not equal to zero if and only if such string str is one of the substrings of S. And Megumi thinks the **value** of such a string is $G(p)$, where $G(x)$ is a polynomial of degree k and p is the length of such a string str, (That is to say $G(x) = a_0 + a_1 * x + a_2 * x^2 + \cdots + a_k * x^k$). More precisely, if such string str is not the substring of string S, its **value** is equal to 0.

It is easy for us to calculate the **value** of any string. To complicate the problem, Megumi will randomly generate another string T whose length is n, and she defines the **power** of string T as the total **value** of all its' substring.

Now Megumi will perform m operations. For each operation, she will put a letter c to the back of string S. After each operation, she wants you to calculate the expected **power** of string T. As the answer may be too large, you should output the answer modulo 998244353.

Input

The first line of the input contains a single integer Q ($1 \le Q \le 10^5$) – the number test cases.

For each test cases, the first line contains four integers l, k, n, m ($1 \le l, m \le 10^5$) ($1 \le n < 998244353$) ($0 \le k \le 50$) – the length of string S, the degree of the polynomial, the length of string T and the number of operations respectively.

The second line contains a string consists only of lower case letters, which stands for the string S.

The third line contains $k+1$ integers and the ith integer is a_i ($1 \le a_i < 998244353$) – the coefficient of x^{i-1} in $G(x)$.

The next m lines contain m lowercase letters and a single letter c per line – the letter Megumi will put to the back of string S for each operation.

It is guaranteed that for all the test cases, $\sum l, \sum m, \sum k \le 3 \times 10^5$.

Output

For each test case, you should output $m+1$ lines.

The first line of your output should be the expected **power** of T without doing any operation to string S.

The next m line of your output should be the expected **power** of T after each operation.

Example

standard input	standard output
1 1 1 2 1 a 1 1 b	921456326 775263736

Problem G. Pangu Separates Heaven and Earth

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

Long long ago, the sky and the earth were not separated, and the universe was chaotic. There was a giant named Pangu who slept for eighteen thousand years in this chaos.

One day, Pangu woke up suddenly. When he saw the darkness around him, he took up his axe and slammed into the darkness in front of him. Just listen to a loud noise, chaotic things gradually separated. Light and clear things rise slowly and become heaven; heavy and turbid things fall slowly and become earth.

After the separation of heaven and earth, Pangu feared that they would still be together. He put his head on the heaven and pedaled the earth with his feet. Every day, Pangu rose by a tenth, and the longer Pangu grew, the higher Pangu grew. In this way, I do not know how many years, the sky and the earth gradually formed, Pangu also fell tired.Pangu's body changed dramatically after his fall. His breath became the wind of the seasons and the floating clouds; his voice turned into a rumbling thunder. His eyes became the sun and the moon; his limbs became the east, west, South and North four poles of the earth; his skin became the vast land; his blood became the running rivers; his sweat became the rain that moistened all things...

Input

The first line has a positive integer T $(1 \le T \le 10)$, which indicates that there are T cases of test data.

Each case of test data input takes up one line and is a positive integer N $(1 \le N \le 100000000)$.

Output

Each case of test data output takes up one line, which is an integer.

When input is 1, output Pangu sleeping years, that is: 18000

When input is others, output zero, that is: 0

Example

standard input	standard output
2 1 2	18000 0

Problem H. The Nth Item

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

For a series F:

$$F(0) = 0, F(1) = 1$$

$$F(n) = 3 * F(n-1) + 2 * F(n-2), (n \geq 2)$$

We have some queries. For each query N, the answer A is the value $F(N)$ modulo 998244353.

Moreover, the input data is given in the form of encryption, only the number of queries Q and the first query N_1 are given. For the others, the query $N_i (2 \leq i \leq Q)$ is defined as the xor of the previous N_{i-1} and the square of the previous answer A_{i-1}. For example, if the first query N_1 is 2, the answer A_1 is 3, then the second query N_2 is $2 \ xor \ (3 * 3) = 11$.

Finally, you don't need to output all the answers for every query, you just need to output the xor of each query's answer $A_1 \ xor \ A_2 ... xor \ A_Q$.

Input

The input contains two integers, Q, N, $1 \leq Q \leq 10^7, 0 \leq N \leq 10^{18}$. Q representing the number of queries and N representing the first query.

Output

An integer representing the final answer.

Example

standard input	standard output
17 473844410	903193081

Problem I. Yukino With Subinterval

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3.5 seconds
Memory limit:	512 megabytes

Yukino has an array $a_1, a_2 \ldots a_n$. As a tsundere girl, Yukino is fond of studying subinterval.

Today, she gives you four integers l, r, x, y, and she is looking for how many different subintervals $[L, R]$ are in the interval $[l, r]$ that meet the following restraints:

1. $a_L = a_{L+1} = \cdots = a_R$, and for any $i \in [L, R]$, $x \leq a_i \leq y$.
2. The length of such a subinterval should be maximum under the first restraint.

Note that two subintervals $[L_1, R_1]$, $[L_2, R_2]$ are different if and only if at least one of the following formulas is true:

1. $L_1 \neq L_2$
2. $R_1 \neq R_2$

Yukino, at the same time, likes making tricks. She will choose two integers pos, v, and she will change a_{pos} to v.

Now, you need to handle the following types of queries:

- 1 pos v: change a_{pos} to v
- 2 l r x y: print the number of legal subintervals in the interval $[l, r]$

Input

The first line of the input contains two integers n, m $(1 \leq n, m \leq 2 \times 10^5)$ – the numbers of the array and the numbers of queries respectively.

The second line of the input contains n integers a_i $(1 \leq a_i \leq n)$.

For the next m line, each containing a query in one of the following queries:

- 1 pos v $(1 \leq pos, v \leq n)$: change a_{pos} to v
- 2 l r x y $(1 \leq l \leq r \leq n)$ $(1 \leq x \leq y \leq n)$: print the number of legal subintervals in the interval $[l, r]$

Output

For each query of the second type, you should output the number of legal subintervals in the interval $[l, r]$.

Example

standard input	standard output
6 3 3 3 1 5 6 5 2 2 3 4 5 1 3 2 2 1 6 1 5	0 4

Note

For the first operations, there are 3 different subintervals($[2, 2]$,$[3, 3]$,$[2, 3]$) in the interval $[2, 3]$, but none of them meets all the restraints.

For the third operations, the legal subintervals in interval $[1, 6]$ are:

$[1, 2], [3, 3], [4, 4], [6, 6]$

Notes that although subintervals $[1, 1]$ and $[2, 2]$ also meet the first restraint, we can extend them to subinterval $[1, 2]$. So the length of them is not long enough, which against the second one.