Problem A. Groups of Permutations

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

Yessine has an array a of n positive integers.

He wonders if he can partition this array into consecutive groups such that every group is a permutation of any size.

For example, if $a = [4, 1, 3, 2, 2, 1, 3, 2, 1]$ he can split it into 3 groups : ($[4, 1, 3, 2]$ $[2, 1]$ $[3, 2, 1]$). However the array $a = [4, 1, 3, 2, 3, 1]$ can't be partitioned into groups of permutations.

A permutation is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[3, 1, 2, 5, 4]$ is a permutation, but $[1, 2, 1]$ is not a permutation (1 appears twice in the array) and $[2, 1, 4]$ is also not a permutation ($n = 3$, but 4 is in the array).

Input

The first line of each test case contains one integer n ($1 \le n \le 2000$) — the size of a

The second line contains n positive integers $a_1, a_2, \ldots,\ a_n$ ($1 \le a_i \le 2000$)

Output

If there is a solution, print "YES". Otherwise, print "NO"

Examples

standard input	standard output

`9`
`4 1 3 2 2 1 3 2 1`
	`YES`

`6`
`4 1 3 2 3 1`
	`NO`

Problem B. Mean Absolute Deviation

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

Rami may loves many branches of mathematics. But, if there is an exception, it must be statistics.

For usual mathematical problems, he will devote his time to understand the structure of the problem and gain intuition, but for statistics, he will forward it to a friend.

But when Statistics and Competitive Programming intersects, that friend should be Yessine who received a recent email from Rami:

`Dear Yessine`

`Today, I have received an interesting problem that I hope that you will like.`

`Spoiler Alert:` `This problem is difficult, even for you ☺:`

`Given` n `reals` $x = [x_1, \ldots, x_n]$, `I challenge you to calculate the mean absolute deviation` $\delta(x)$ `(MAD):`

$$\delta(x) = \frac{1}{n}\sum_{i=1}^{n}|x_i - \mu(x)|$$

$$\texttt{where: } \mu(x) = \frac{1}{n}\sum_{i=1}^{n} x_i \texttt{ is the mean}$$

`Now, to make things worse, I challenge you to redo the calculation for` q `subarrays of` $x : x_{[l_1,r_1]}, \ldots, x_{[l_q,r_q]}$`:`

$$\texttt{Find } \delta(x_{[l_i,r_i]}) \texttt{ for each } i \in \{1, \ldots, q\}$$

`Here is the list of constraints:`

$1 \le n \le 10^5$	$1 \le q \le 10^5$	$1 \le l_i \le r_i \le n$

`Best Regards,`

`Rami.`

Yessine was mad that such an easy problem was given to him, or so he thought until reading the constraints.

Now, neither Rami nor Yessine are capable of solving this problem, so they both request your help.

Input

1. The first line contains two integers $1 \le n, q \le 10^5$ representing the size of the array and the number of queries

2. The second line contains n reals $x_1, \ldots, x_n$ representing the values of the array

3. Each of the following q lines contains 2 integers l, r with $1 \le l, r \le n$ representing the considered subarray $x_{[l,r]}$

Output

q reals. with the ith line representing the mean absolute deviation of the subarray $x_{[l_i,r_i]}$

Examples

standard input	standard output

5 7
0 10 0 1 3
1 3
2 4
2 2
4 5
3 5
1 2
3 4
	4.44444
	4.22222
	0
	1
	1.11111
	5
	0.5

5 7
4 3 0 5 8
3 4
2 2
1 4
3 5
2 3
2 3
3 4
	2.5
	0
	1.5
	2.88889
	1.5
	1.5
	2.5

Note

Yessine received a clarifying mail from Rami:

To Yessine

Well, I accidently omitted the definition of the mean absolute deviation of a subarray, here is it with a little example:

- The mean of the subarray $x_{[l_i,r_i]}$ is :

$$\mu(x_{[l_i,r_i]}) = \frac{1}{r_i - l_i + 1}\sum_{j=l_i}^{r_i} x_j$$

- The mean absolute deviation of the subarray $x_{[l_i,r_i]}$ is :

$$\delta(x_{[l_i,r_i]}) = \frac{1}{r_i - l_i + 1}\sum_{j=l_i}^{r_i} |x_j - \mu(x_{[l_i,r_i]})|$$

- For a little example, consider the array $x = [0, 10, 0, 1, 3]$ and the query $l = 1, r = 3$. We have $x_{[1,3]} = [0, 10, 0]$. The mean of the subarray is $\mu(x_{[1,3]}) = \frac{10}{3}$ and the MAD is:

$$\delta(x_{[1,3]}) = \frac{1}{3}\sum_{j=1}^{3} |x_j - \mu(x_{[1,3]})| = \frac{|0 - \frac{10}{3}| + |10 - \frac{10}{3}| + |0 - \frac{10}{3}|}{3} = \frac{\frac{10}{3} + \frac{20}{3} + \frac{10}{3}}{3} = \frac{40}{9} \approx 4.44444$$

Best Regards,

Rami.

Problem C. Decrypting the Password

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	256 megabytes

Yessine was trying to log into his email when he realized that he forgot his password. Fortunately, he wrote it in a text file and saved it just in case. However, when he opened the file, he didn't find the password but rather a very long string of digits. He then remembered that he didn't write the password but rather an encryption of it. As a matter of fact, the password is the number of substrings divisible by 11 in the string. Yessine wants to open his email, help him decrypt the password.

A string a is a substring of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the beginning and several (possibly, zero or all) characters from the end.

Input

The first line contains one integer t $(1 \le t \le 1000)$ — the number of test cases.

The first line of each test case contains one integer n $(1 \le n \le 10^6)$ — the length of the string s.

The second line of each test case contains a string consisting of n decimal digits.

It is guaranteed that the sum of n over all test cases does not exceed 10^6

Output

Print one line containing the answer — The number of substrings divisible by 11

Example

standard input	standard output
4 3 121 4 1111 6 123456 8 20654301	1 4 0 3

Note

In the first case, the only substring divisible by 11 is 121

Problem D1. Rami's Scheme (Easy Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	1024 megabytes

This is the easy version of the problem. The difference between the two versions are the constraints.

Rami always was fond of random numbers, he always wonders how randomness arises from the deterministic nature of mathematics.

Wanting to impress his friends, he created a new pseudo-random number generation scheme, that he proudly called Rami Scheme

A Rami Scheme consists of the following steps:

1. choose 4 integer parameters: p, a, b such that $0 \leq a, b, < p$ with p prime

2. choose 2 seeds u_0, u_1

3. for $k > 1$, u_k will be generated with the following rule:

$$u_k = (au_{k-1} + bu_{k-2}) \bmod p$$

4. using the rule above, he will calculate many such numbers and use them to generate the following random numbers $(v_k)_{k\in\mathbb{N}}$:

$$v_k = \left(\sum_{i=0}^{k} iu_i\right) \bmod p$$

5. Finally, after calculating many terms $v_0, \ldots, v_{10^{18}}$, he will choose s numbers $v_{n_1}, \ldots, v_{n_s}$. those final numbers will be his true random numbers.

Rami wants you to test the robustness of this scheme, so he asks you for help.

- First of all, he wants you to measure the robustness index R of this scheme, which is defined as the eventual fundamental period of the sequence $(v_k)_{k\in\mathbb{R}}$. In other words,he wants the smallest strictly positive integer R such that:

$$\exists N \in \mathbb{N}/ \quad \forall k \in \mathbb{N}_{\geq N}, v_{k+R} = v_k$$

- After that, he knows that he cannot calculate all terms of the sequence $(v_k)_{k\in\mathbb{N}}$, and he only needs s terms $v_{n_1}, \ldots, v_{n_s}$ of the sequence. So he asks your help for it.

As the number R may be too big, Rami will be happy if you only give him $R \bmod 2^{64}$.

Input

The first line contains 6 integers, p, a, b, u_0, u_1, s:

- p, a, b : the parameters of the scheme, $0 \leq a, b < p < 100$, with p a prime number

- u_0, u_1 : the seeds, $0 \leq u_0, u_1 < p$

- s : the number of terms to calculate, $0 \leq s \leq 10^6$

The second line contains s integers, $n_1, \ldots, n_s$ representing the terms to calculate. $0 \leq n_i \leq 10^6$

Output

The first line contains one integer R : the robustness index of the selected scheme, modulo 2^{64}.

The second line contains s integers $v_{n_1}, \ldots, v_{n_s}$: the calculated terms.

Examples

standard input	standard output

97 3 5
0 1
3
2 7 3
	912576
	7 79 49

17 0 0
2 3
7
0 1 2 3 4 5 6
	1
	0 3 3 3 3 3 3

Note

- In this version, it's guaranteed that $R < 10^6$ - It is also guaranteed that the sequence $(v_n)_{n\in\mathbb{N}}$ is eventually periodic

Problem D2. Rami's Scheme (Hard Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	1024 megabytes

This is the hard version of the problem. The difference between the two versions are the constraints.

Rami always was fond of random numbers, he always wonders how randomness arises from the deterministic nature of mathematics.

Wanting to impress his friends, he created a new pseudo-random number generation scheme, that he proudly called Rami Scheme

A Rami Scheme consists of the following steps:

1. choose 4 integer parameters: p, a, b such that $0 \leq a, b, < p$ with p prime

2. choose 2 seeds u_0, u_1

3. for $k > 1$, u_k will be generated with the following rule:

$$u_k = (au_{k-1} + bu_{k-2}) \bmod p$$

4. using the rule above, he will calculate many such numbers and use them to generate the following random numbers $(v_k)_{k \in \mathbb{N}}$:

$$v_k = \left(\sum_{i=0}^{k} iu_i\right) \bmod p$$

5. Finally, after calculating many terms $v_0, \ldots, v_{10^{18}}$, he will choose s numbers $v_{n_1}, \ldots, v_{n_s}$. those final numbers will be his true random numbers.

Rami wants you to test the robustness of this scheme, so he asks you for help.

- First of all, he wants you to measure the robustness index R of this scheme, which is defined as the eventual fundamental period of the sequence $(v_k)_{k \in \mathbb{R}}$. In other words,he wants the smallest strictly positive integer R such that:

$$\exists N \in \mathbb{N} / \quad \forall k \in \mathbb{N}_{\geq N}, v_{k+R} = v_k$$

- After that, he knows that he cannot calculate all terms of the sequence $(v_k)_{k \in \mathbb{N}}$, and he only needs s terms $v_{n_1}, \ldots, v_{n_s}$ of the sequence. So he asks your help for it.

As the number R may be too big, Rami will be happy if you only give him $R \bmod 2^{64}$.

Input

The first line contains 6 integers, p, a, b, u_0, u_1, s:

- m, a, b : the parameters of the scheme, $0 \leq a, b < p < 10^9$, with p a prime number.

- u_0, u_1 : the seeds, $0 \leq u_0, u_1 < p$

- s : the number of terms to calculate, $0 \leq s \leq 10^6$

The second line contains s integers, $n_1, \ldots, n_s$ representing the terms to calculate. $0 \leq n_i \leq 10^{18}$

Output

The first line contains one integer R : the robustness index of the selected scheme, modulo 2^{64}.

The second line contains s integers $v_{n_1}, \ldots, v_{n_s}$: the calculated terms.

Examples

standard input	standard output

7 2 6
0 1
14
0 1 2 3 4 5 6 7 8 9 10 11 12 13
	7
	0 1 5 0 2 6 0 0 1 5 0 2 6 0

499 13 41
0 15
5
0 1 2 3 4
	31062750
	0 15 405 374 47

975151579 1 1
0 1
5
0 1 2 3 4 5
	950920601051041662
	0 1 3 9 21

Note

- A prime number is a number whose only divisors are 1 and itself
- A period T of a sequence $(H_n)_{n\in\mathbb{N}}$ is a strictly positive integer such that $\forall k \in \mathbb{N}, \quad H_{T+k} = H_k$.
- A sequence is said to be periodic if it has at least one period.
- The fundamental period of a periodic sequence is its smallest period.
- It is guaranteed that the sequence $(v_n)_{n\in\mathbb{N}}$ is eventually periodic
- For the first test case, the first 14 terms of the sequence $(u_n)_{n\in\mathbb{N}}$ are: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13$. Therefore, the first 14 terms of the sequence $(v_n)_{n\in\mathbb{N}}$ would be: $0, 1, 5, 0, 2, 6, 0, 0, 1, 5, 0, 2, 6, 0$. It can be proven that $(v_n)_{n\in\mathbb{N}}$ is indeed periodic as it appears, and its fundamental period is 7.
- For the second test case, the first 5 terms of the sequence $(u_n)_{n\in\mathbb{N}}$ are: $0, 15, 195, 186, 324$. Clearly, $v_0 = 0,\ v_1 = 15$, we have: $v_2 = (15 + 2 \cdot 195) \bmod 499 = 405, \quad v_3 = (405 + 3 \cdot 186) \bmod 499 = 464$, and $v_4 = (464+4\cdot 464) \bmod 499 = 263$. It can be proven that $(v_n)_{n\in\mathbb{N}}$ is periodic with a fundamental period of 124251000.

Problem E. 1D Monopoly

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

Rami loves monopoly

> Insert Story here

A **1D Monopoly** array is an array $A = (A_1, \dots, A_n)$ filled with the cost of each house.

The game can be played with any number of players, but in this case **Rami** chose to investigate the game with only one player:

1. The game starts with the player at position 0 having M units of money and 0 houses.

2. The player will throw a dice, and based on its value k, will advance k units.

3. After advancing by k units, One of two scenarios will happen: 1. If the position of the player exceeds n, than the game is finished 2. Else, the player will be located at a position $0 < s \leq n$, he will try to buy the house at that position s, Note that he will succeed only if his remaining money is **greater than or equal** to A_s. In that case, his remaining money will be updated to $M - A_s$, and the number of his houses will increase by 1.

4. If the game is not finished, repeat step 2.

At the start of the game, **Rami** has M units of money, and he knows the cost of each house, and that his dice is **fair**. He wants to calculate the expected number of houses that he will possess at the end of the game.

Input

1. The first line contains 2 integers: n, M where:

- $1 \leq n \leq 1000$: the size of the array A

- $1 \leq M \leq 1000$: the initial amount of money

2. The second line contains n integers $0 \leq A_1, \dots, A_n \leq 10^6$, with the i^{th} integer represents the cost of the house at position i

Output

One integer E : the expected number of houses that **Rami** will possess at the end of the game

Examples

standard input	standard output

`2 2`
`1 1`
	`0.361111`

`6 6`
`1 1 1 1 1 1`
	`1.52163`

`2 1`
`1 1`
	`0.333333`

Problem F. Journey Through Time

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

Oussama is a student. Next year will be his last one at university so he will need to find a graduation internship. He wants to find an internship abroad but he is afraid he won't be able to travel due to the wide spread of COVID19, so he started thinking about a way to end the pandemic. Suddenly, a very good idea hit him : he will make a time machine and go back in time to find the very first infected persons and save them from the infection thus preventing the pandemic. However, time travel is dangerous and can cause some serious damage.

You might have heard that the timeline is linear, but it is not. In fact, the timeline is a tree of N nodes, where each node is a different time. Each node i causes a certain damage D_i.

There are M special nodes. Each special node denotes a time of an infection that should be stopped. Oussama is initially at the present, which is node 1, and will visit each one of the special nodes in any order he wants. Whenever he moves from a node A to a node B, he will get a damage equal to the maximum node damage on the path between A and B. However, he won't get any more damage from the nodes on that path after the first time he travels through it. In other words, after he walks through a path and takes the damage, the damage of the nodes on that path become zeros. When he visits the node he wants, he stops to prevent the infection from happening and then travel from there to one of the remaining special nodes. He repeats the same process until every special node is visited . The total damage he will get is the sum of the damages obtained on each move.

Oussama wants to save the world from the pandemic but he has a limited stamina so he can't take too much damage. Help him determine the minimal damage he will take during the time travel .

Input

The first line contains 2 integers N and M $(1 \leq M \leq N \leq 10^5)$

The second line contains M space separated integers denoting the indices of special nodes.

The third line contains N space separated integers denoting the damage of each node $(0 \leq d_i \leq 10^9)$.

Finally, there are $N - 1$ lines, each line contains 2 space separated integers U V $(1 \leq U, V \leq N)$ which indicates that there is an edge between nodes U and V.

Output

Print a single integer, the minimal total damage.

Examples

standard input	standard output

10 4
3 6 8 9
0 5 20 50 5 7 8 15 1 50
1 2
1 9
9 10
2 3
3 4
3 5
5 6
5 8
6 7
	28

10 5
4 6 7 9 10
0 1 1 3 8 5 4 11 9 10
1 2
1 3
2 4
2 5
5 9
5 10
3 6
6 7
7 8
	27

Problem G. Non-Increasing Dilemma

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

You are given an array $a_1, a_2, \ldots, a_n$.

You have to perform this operation on the array any number of times :

- Choose an integer i $(1 \le i \le n)$ and for all $j \ne i$ add $a[i]$ to $a[j]$

What is the minimum number of operations needed to make the array sorted in non-increasing order?

Input

Each test contains multiple test cases. The first line contains a single integer t $(1 \le t \le 10^5)$ — the number of test cases. Description of the test cases follows.

The first line of each test case contains a single integer n $(1 \le n \le 10^5)$ — the length of the array.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$ — the elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each test case, print one integer k — The minimum number of operations needed to make the array sorted in non-increasing order.

Example

standard input	standard output

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

Note

In the first test case, in the first operation, we choose $i = 3$, the array becomes $[6, 3, 2, 5]$. In the second operation, we to choose $i = 4$, the array becomes $[11, 8, 7, 5]$

Problem H. Two Shortest Paths

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

Yessine and Rami are living in Sfax.

Sfax is a rectangular grid of size $n * m$. Each cell has a number written on it, the number on the cell (i, j) is $a_{i\ j}$.

Yessine is living in cell $(1, 1)$ and wants to go to the cell (n, m).

Rami is living in cell $(1, m)$ and wants to go to the cell $(n, 1)$.

Rami and Yessine can move to any other cell cell that share the same edge with their current cell, in other words they can move **UP**, **DOWN**, **LEFT**, or **RIGHT**.

Between all the paths, to reach their destinations, Yessine and Rami take the path that has the minimum sum path including the start cell and the end cell.

The sum path of some path is the sum of all numbers in all cells in this path.

As Oussama is their best friend, he wanted to put on each cell (i, j) **distinct** positive integers so that the sum of the two shortest paths of Yessine and Rami is as minimal as possible.

Oussama is stuck. Please help him determine what is the minimal sum of the two shortest paths.

Input

The first line contains a single integer t $(1 \leq t \leq 10000)$ — the number of test cases.

Each test case contains a line of two integers n and m denoting the size of the grid $(2 \leq n, m \leq 10^4)$

Output

For each test case, print a single integer k — The minimal sum of the two shortest paths.

Example

standard input	standard output

2
5 5
7 3
	106
	94

Problem I. Simulation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Rami and **Oussama** frequently talk about simulations, and how on many occasions, order arises from disorder during many simualtions.

To test this observation, **Oussama** conceived the following simulation:

Initially, he will have an array A of size n **filled with zeros**. Then, he will do m times these two steps:

1. Select a number k uniformly in $\{1, \dots n\}$

2. Increment A_k by 1

Upon investigating the simulation, he noted that sometimes, there is atleast one element which grows too much. So he wanted to know how likely such an event occurs by doing the following:

1. He chose the size of the array n, and a threshold K.

2. He applied m steps of the simulation on this array, and memorized the content of the array at the final state

3. Finally, He masked the content of the array, and invited **Rami** to guess the probability that **there exists at least** one element of A greater or equal than the chosen threshold K.

As the question is a little bit hard, **Oussama** wanted to ease it and gave **Rami** the value of the first s elements of $A : A_1, \dots, A_s$.

Now, **Rami** was given a simulation of m steps on an array A of size n and a threshold K, and he knows the values of $A_1, \dots, A_s$. He must correctly guess the probability that there exists an element of A greater or equal to K.

Rami is completely stuck, so he asked your help about it.

Input

1. The first line contains 4 integers: n, m, K, s where: - $1 \leq n \leq 300$: the size of the array A.

- $1 \leq m \leq 300$: the number of steps of the simulation.

- $0 \leq K \leq 10^6$: the threshold.

- $0 \leq s \leq n$: number of given elements.

2. The second line contains s integers $0 \leq A_1, \dots, A_s, \leq m$

Output

a real $0 \leq p \leq 1$ the probability that all elements of A are less than K after m steps of the simulation, while knowing the values of $A_1, \dots, A_s$

Examples

standard input	standard output
`2 10 6 0`	`0.753906`
`300 300 5 0`	`0.671265`
`10 3 2 0`	`0.28`

Note

- **Rami** doesn't know the state of the array at anytime (except the initial state), he only knows the size n, the number of iterations m, the threshold K, and the final values $A_1, \dots, A_s$ of the first s elemnts.

- For the first test case, we have a simulation with an array of size 2, and 10 iterations. The threshold is $K \geq 6$, the only case when $\max(A_1, A_2) < 6$ is when $A_1 = A_2 = 5$. This comes with a probability:

$$p = 1 - \binom{10}{5} \times \frac{1}{2^{10}} = 1 - \frac{10!}{(5!)^2 2^{10}} = \frac{193}{256} \approx 0.75390625$$

Where $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ is the binomial coefficient