Problem A. Legendary Numbers

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`White`

Sanad made a club for some specific numbers and he called them **Legendary Numbers**.

A Legendary Number is a number whose prime factors are limited to $2, 7, 11, 17$.

For example: $2, 68, 1, 7, 8, 14, 11, 17$ are Legendary Numbers, but $3, 27, 25, 61$ are **not** Legendary Numbers.

Sanad will give you two numbers X, Y — he wants you to print the summation of all Legendary Numbers from X to Y **inclusive** modulo $10^9 + 7$.

Input

The input consists of multiple test cases. The first line contains a single integer $T (1 \leq T \leq 10^5)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers X and Y $(1 \leq X \leq Y \leq 10^{17})$.

Output

For each test case, print one integer — the summation of all Legendary Numbers from X to Y **inclusive** modulo $10^9 + 7$.

Example

standard input	standard output
5 1 10 3 77 5 350 125 287 6 61	22 551 4234 2209 338

Problem B. Who is the winner

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Black`

In the lively city of Alexandria, Nasr and 7oskaa playfully competed to find out who was taller. However, accurately measuring their heights proved to be a tricky task. Nasr's height was represented by A, and 7oskaa's height was represented by B.

To settle the debate, the city organized an exciting competition with various challenges to measuring their heights, aiming to solve this mystery once and for all.

Now, it's up to you to determine if $Nasr$ is the winner (taller) individual or not and put an end to the suspense.

Input

Only one line contains two space-separated integers A, B $(1 \leq A, B \leq 100)$ — Nasr's height followed by 7oskaa's height.

Output

Print "YES" if Nasr is the winner, otherwise print "NO".

Examples

standard input	standard output
2 3	NO
3 2	YES
2 2	NO

Problem C. Beauty Subarray

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Red`

You are given an array A consisting of N integers. The beauty of the array is the maximum *sum* of some **consecutive subarray** of this array (this subarray may be empty). For example, the beauty of the array $[10, -5, 10, -5, 10]$ is 20, and the beauty of the array $[0, -1, 0]$ is 0.

You may choose **at most one** consecutive subarray of A and divide **all values** contained in this subarray by X or multiply **all values** contained in this subarray by X. You want to maximize the beauty of the array after applying at most one such operation (you can do zero or one operation).

note:

- if the value is 5 and you will divide it by 2 then the new value will be 2
- if the value is -5 and you will divide it by 2 then the new value will be -2

Input

The first line of input contains an integer T — the number of test cases.

The first line of each test case contains two integers N and X $(1 \le N \le 3 \cdot 10^5, 1 \le X \le 100)$ — the length of the array A and the integer X respectively.

The second line contains N integers $A_1, A_2, ..., A_N$ $(-10^9 \le a_i \le 10^9)$ — the array A.

Output

Print one integer — the maximum possible beauty of array A after dividing all values belonging to some consecutive subarray by x or multiplying all values belonging to some consecutive subarray by x.

Example

standard input	standard output
2 5 1 10 5 10 5 10 5 3 10 -5 10 -5 10	40 60

Note

In the first test case, one of the best solutions is not to apply any operation and to take the whole array as a beauty subarray, so the answer is 40.

In the second test case, the best solution is to multiply all values of the array by x and then take the whole array as a beauty subarray, so the answer is 60.

Problem D. Merge The Machines

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Pink`

Nasr, an avid machine enthusiast, has invited you to his house to put your skills to the test. He has provided you with an array A of n machines, each with its own power level denoted by A_i.

Your task is to merge all the machines into a single, powerful machine. Here's how it works:

- Choose a center machine, denoted by index K.
- The power of the center machine remains unchanged.
- Machines to the left of the center lose 2 units of power each, i.e. $A_i = A_i - 2$ for all $(1 \le i < K)$.
- Machines to the right of the center lose 3 units of power each i.e. $A_i = A_i - 3$ for all $(K < i \le N)$.
- The total power of the merged machine is equal to the sum of all new powers.
- If a machine's power becomes negative during this process, it will be counted as having zero power.

Your goal is to determine the maximum possible power that can be achieved through this merging process.

Input

The first line contains a single integer T — the number of test cases.

Each test case consists of two lines:

1. A single integer N $(1 \le N \le 10^6)$ — the number of machines.
2. N space-separated integers $A_1, A_2, \ldots, A_N$ $(1 \le A_i \le 10^9)$ — where A_i denotes the power of the i_{th} machine.

Output

For each test case, print a single integer — the maximum power we can get after merging the machines.

Example

standard input	standard output
3 5 1 3 4 2 5 4 1 3 1 1 5 7 8 2 1 2	8 3 13

Note

In the first test case, we can choose the 5_{th} machine as a center machine, so the new powers will look like this $A = [0, 1, 2, 0, 5]$, after merging the powers we will get a machine with 8 power.

Problem E. Sanad And Chicks

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Yellow`

Sanad got bored of programming so, he decided to start a new business (chicks business) , he is going to buy some chicks and feed them till they grow and become chickens.

Sanad bought N chicks each having a weight W_i.

Sanad wonders how many ways he can arrange the chicks so that there is a chick with prime weight at each **even** index and as the number of ways may be too large print it modulo $10^9 + 7$.

Note: Sanad starts counting from 1 not 0.

Input

The first line contains an integer number T — the number of test cases.

The first line of each test case contains one number N $(2 \leq N \leq 10^6)$ — the number of chicks.

The second line of each test case contains N numbers W_i $(1 \leq W_i \leq 10^7)$ — the weight of the i_{th} chick.

Output

Print the number of ways that Sanad can arrange the chicks so that at each **even** index there is a chick with prime weight modulo $10^9 + 7$.

Remember: Sanad start counting from 1.

Example

standard input	standard output
2 4 10 11 17 4 4 11 1 7 7	4 12

Note

Valid ways to arrange the chicks in test case 1:

[4, <u>11</u>, 10, <u>17</u>], [4, <u>17</u>, 10, <u>11</u>], [10, <u>11</u>, 4, <u>17</u>], [10, <u>17</u>, 4, <u>11</u>]

By looking at these 4 ways of arranging the chicks, you will find that at each **even** index there is a chick with prime weight.

Problem F. Subarrays

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark gold`

Given four number N, P, Q and R. Count the number of sequences A of length N which have four indices X, Y, Z, W ($0 \le X < Y < Z < W \le N$) that satisfy the following conditions:

- $A_X + A_{X+1} + \cdots + A_{Y-1} = P$
- $A_Y + A_{Y+1} + \cdots + A_{Z-1} = Q$
- $A_Z + A_{Z+1} + \cdots + A_{W-1} = R$
- $1 \le A_i \le 15$ for all i ($0 \le i < N$)

In other words count the number of sequences of length n that contain three consecutive subarrays such that their sum is P, Q, and R respectively and the value of each element in the sequence is between 1 and 15 inclusive.

Since the number can be very large, Print it modulo $(10^9 + 7)$.

Input

The only line of input contains N, P, Q and R ($3 \le N \le 50$, $1 \le P, Q, R \le 14$, $3 \le P+Q+R \le 16$).

Output

Print the answer modulo $(10^9 + 7)$.

Examples

standard input	standard output
`3 1 2 3`	`1`
`4 3 1 4`	`35`

Note

In the first example the only valid sequence is $[1, 2, 3]$

In the second example some of the valid sequences are: $[1, 2, 1, 4]$, $[7, 3, 1, 4]$, $[3, 1, 4, 10]$, $[3, 1, 2, 2]$.

Problem G. Escaping From Prison

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Blue-purple`

Nasr was caught stealing problems from other communities and was taken to prison, where the rooms are laid out in an $N \times N$ grid.

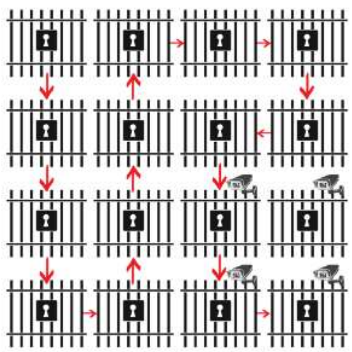
Nasr begins in the first room on the top left $(1, 1)$, and there is a hole in room (N, N) through which Nasr can escape the prison.

All rooms in the subrectangle starting from (R, C) and ending with (N, N) are under camera surveillance for the first K seconds when Nasr begins moving from his room.

It is guaranteed that room $(1, 1)$ is not under camera surveillance.

In one second, Nasr can move from the current room (X_1, Y_1) to room (X_2, Y_2) if and only if the condition $|X_2 - X_1| + |Y_2 - Y_1| = 1$ is satisfied.

Note that Nasr should change his position every second, **and one room can't be visited more than once**.



example one (test case 1)

In the example given above, Nasr's first visit to a room that is under camera surveillance at a second of 12 so if $(K \geq 12)$ Nasr will get caught by the cameras.

Can Nasr escape the prison without getting caught by the cameras?

Input

The first line contains a single integer T $(1 \leq T \leq 100)$ — the number of test cases.

Each test consists of a single line containing four space-separated integers N, R, C, K $(2 \leq N \leq 10^9)$ $(1 \leq R, C \leq N)$ $(1 \leq K \leq 10^{18})$ — the size of the prison, the top left corner of the sub rectangle that is under camera surveillance, and the duration for the camera to shut down.

Output

Print **"YES"** if Nasr can escape the prison, and **"NO"** otherwise.

You can print "YES" and "NO" in any (lower or upper) case.

Example

standard input	standard output
2 4 3 3 11 5 3 4 22	YES NO

Problem H. Frog Jumps

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark grey`

Given two integers n and k and an array a of k integers.

A frog is standing at point 0 on the OX axis. It can jump from its current position i to position $i + j$ for any $j \in [1, k]$ with cost a_j.

Find the total sum of costs of **all possible paths** the frog can take to go from point 0 to point n modulus $10^9 + 7$.

The cost of a path is the sum of the costs of all jumps the frog has done on this path.

Input

The first line contains two integers n, k ($1 \le n \le 10^9$, $1 \le k \le 100$).

The second line contains k integers $a_1, a_2, \ldots, a_k$ ($1 \le a_i \le 10^9$).

Output

For each test case output one integer the total sum of costs of **all possible paths** the frog can take to go from point 0 to point n modulus $10^9 + 7$.

Example

standard input	standard output
5 3 1 2 3	65

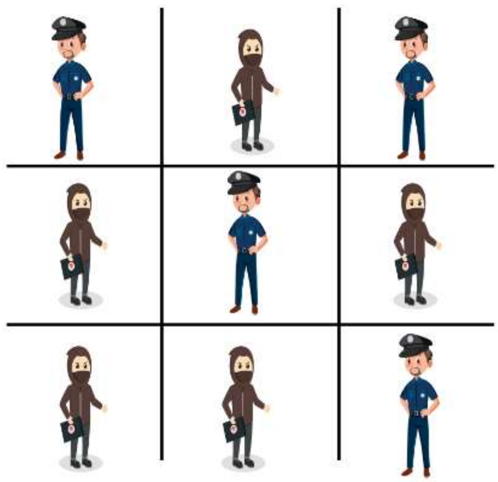
Problem I. MeMo And The Traps

Input file:	standard input
Output file:	standard output
Balloon Color:	Blue

MeMo decided to shift his career, he is going to work with policemen.

As MeMo was a programmer, The policemen asked MeMo to make a program that tells them the number of thieves they can catch, knowing their positions.

A grid of size $N \times N$ holds P *and* T the policemen's positions and thieves' positions (i.e. each cell in the grid contains either a policeman or a thief).



- A policeman can catch a thief just in his row.
- A policeman cannot catch a thief who is more than K units away from him.

Now because MeMo left his career, he asks you to help him and tell him the number of thieves the policemen can catch.

Input

The first line of input contains two integers N and K ($1 \le N, K \le 3000$), the size of the grid, and the maximum distance between a thief and police such that the police can catch the thief.

The next N lines of input contain the grid of P and T which represent the policemen's positions and thieves' positions respectively.

Output

Your task is to print the maximum number of thieves that can be caught.

Example

standard input	standard output
3 1 P T P T P T T T P	4

Problem J. Mod X

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Sky blue`

MeMo gives you three integer numbers N, K, and Q $(1 \le K \le 2)$ the size of the array, the type of queries, and the number of queries respectively. Then you are given an array A with that length N of integer values, and each query you are given L, R $(1 \le L \le R \le N)$.

- If K equals 1, you can concatenate the values of indices from L to R and print it using modulo X.
- If K equal 2, you can concatenate the **odd** values of indices from L to R and print it taking modulo X.

So your task is for each query you have to print the concatenated value modulo X.

Input

The first line of input contains an integer T — the number of test cases.

The first line of each test case contains four integers N, K, X, Q $(1 \le N \le 2 \cdot 10^5)$, $(1 \le K \le 2)$, $(1 \le X \le 10^9)$, $(1 \le Q \le 2 \cdot 10^5)$, the length of the array, the type of queries, the value of X, and the number of queries respectively.

The second line of each test case contains the array A, $(0 \le A_i \le 10^9)$.

Then you have q lines of queries, each line contains two integers L, R $(1 \le L \le R \le N)$ the range of indices of the array.

Output

Your task is for each query you have to print the concatenated value modulo x.

Example

standard input	standard output
3 2 1 3 1 3 1 1 2 3 2 7 2 4 2 1 1 3 2 3 6 2 1 4 1 2 3 4 5 6 1 3 1 4 2 4 2 5	1 6 2 0 0 0 0

Note

Second test case explanation:

The query of type 2 (i.e. odd indices required).

In the first query, $L = 1$, $R = 3$ then the concatenated value is 41 when taking modulo 7 the result is 6.

In the second query, $L = 2$, $R = 3$ then the concatenated value is 2, when taking modulo 7 the result is 2.

Problem K. A Perfect Picture

Input file:	standard input
Output file:	standard output
Balloon Color:	Navy blue

Nasr should take as many *perfect pictures* of his students as possible, he has N students, and the height of the i_{th} student is A_i.

A picture of a group of students is called **perfect** if and only if it satisfies the following conditions:

- The group of students is arranged in two rows (back and front), and each row must contain at least one student.
- The height of any student in the back row is strictly higher than all the students in the front row.

Nasr can't take a picture of the same group of students more than once (even if he changes arrangement), e.g., if Nasr took a *perfect picture* like this, front row $\rightarrow [A_1, A_2]$, back row $\rightarrow [A_3]$, he can't take another picture of the same students like this, front row $\rightarrow [a_1]$, back row $\rightarrow [A_2, A_3]$ because they are the exact same students.

Calculate the maximum *perfect pictures* Nasr can take of his students, and since the answers may be large, print them modulo $10^9 + 7$.

Input

The first line contains a single integer T — the number of test cases.

The first line of each test case contains a single integer N $(1 \leq N \leq 10^5)$ — the number of students.

The second line of each test case contains N space-separated integers $A_1, A_2, \ldots, A_n$ $(1 \leq A_i \leq 10^9)$ — where A_i denotes the height of the i_{th} student.

Output

For each test, Print a single integer — the maximum *perfect pictures* Nasr can take of his students modulo $10^9 + 7$.

Example

standard input	standard output
3 4 5 6 3 2 3 3 5 3 2 1 2	11 3 1

Note

In the second test case, when the array is 1−indexed,

Nasr can take the perfect pictures as follows:

1. front row $\rightarrow [a_1, a_3]$, back row $\rightarrow [a_2]$.
2. front row $\rightarrow [a_1]$, back row $\rightarrow [a_2]$.
3. front row $\rightarrow [a_3]$, back row $\rightarrow [a_2]$.

Problem L. Exhausted

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Grey`

Omar and his team are attending ECPC. They are tired of solving problems and should solve more to be qualified. Omar brings N magical candies (it's not an electronic device).

Each candy increases his power by A_i units, and there are M problems left he needs to solve, each of them needs B_i units of power to be solved.

Omar wants to know the **minimum** power he needs from eating candies to solve each problem.

Note: the operations are independent.

Input

The first line contains integer N, M $(1 \leq N, M \leq 10^5)$.

The second line contains n integers a_i $(1 \leq A_i \leq 10^4)$.

The third line contains m integers B_i the number of problems $(1 \leq B_i \leq 10^5)$.

Output

print M integers, the i_{th} integer is the minimum power Omar needs to gain to solve the i_{th} problem. if he can't solve this problem print -1

Examples

standard input	standard output

`5 3`
`1 2 3 4 5`
`3 8 16`
	`3 8 -1`

`5 6`
`6 8 2 1 5`
`1 4 3 11 18 13`
	`1 5 3 11 19 13`

Note

In the first sample test:

To solve the first problem : Omar can eat candies with power [1, 2].

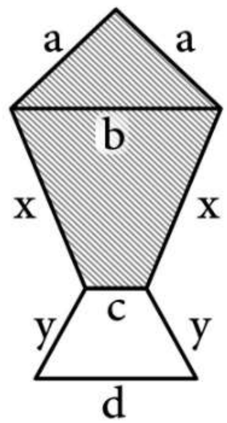
To solve the second problem : Omar can eat candies with power [1 , 3 , 4].

To solve the third problem : Omar can't solve this problem even if he eats all candies.

Problem M. Ramadan Lantern

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Red-purple`

Sanad wants to prepare for Ramadan right now, so he will make a flat wooden lantern.



The lantern Sanad is designing right now.

Now Sanad is in the designing phase, and he wants to make a design such that, he wants to paint the lantern with two different colors as shown in the above image, he wants the difference between the quantity of the two paintings to be as small as possible.

He can just control the position of the line segment equal to C, he can slide it up or down, changing the values of X and Y.

We can say that: $E = X + Y$

Given A, B, C, D, E, you have to print the value of X, such that it makes the quantity of the two paintings as **convergent** as possible.

Input

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

In each test case, you will be given five integer numbers A, B, C, D, E $(1 \leq A, B, C, D \leq 1000)$, $(1 \leq E \leq 2000)$, $(B < 2 \times a)$, $(C \leq B)$, $(C \leq D)$.

It is guaranteed that we can make a valid lantern using the given constraints.

Output

In each test case, you have to print only one number X, such that it makes the difference between the quantity of the two paintings to be as small as possible.

The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}.

Example

standard input	standard output

`2`
`12 20 10 14 40`
`123 50 30 42 302`
	`15.737328`
	`103.648153`

Problem N. Xor Range

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

You are given N integers and Q queries.

Each query is given integer X and need to compute the value, $\underset{Y \in A_L, \ldots, A_R}{\mathtt{max}(X \otimes Y)}$, i.e. the maximum value of bitwise exclusive OR (also known as XOR) of integer X and some integer Y from the subarray $A_L, A_{L+1}, \ldots, A_{R-1}, A_R$.

Input

The first line contains two integer N, Q ($1 \leq N, Q \leq 2 \times 10^5$) — N is the number of integer are given and Q is the number of query.

The next line contains N integer ($0 \leq A_i < 2^{20}$) — A_i denote the i_{th} element of A.

The next Q lines, each line contains three integer L, R, X ($1 \leq L \leq R \leq N, 0 \leq X < 2^{20}$).

Output

For each query print the value, $\underset{Y \in A_L, \ldots, A_R}{\mathtt{max}(X \otimes Y)}$, i.e. the maximum value of bitwise exclusive OR (also known as XOR) of integer X and some integer Y from the subarray $A_L, A_{L+1}, \ldots, A_{R-1}, A_R$.

Example

standard input	standard output
6 4 1 3 4 9 5 2 1 4 3 2 5 0 4 6 11 1 6 8	10 9 14 13

Problem O. Count Nodes

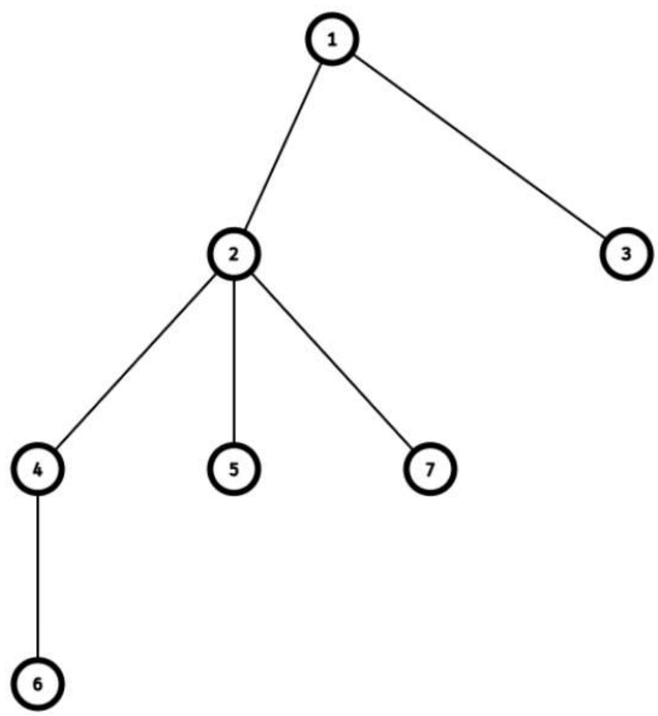
Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Gold`

You are given a rooted tree with N vertices, numbered $1, 2, \ldots, N$. Vertex 1 is the root.

There are Q queries, each consisting of integers U_i and D_i. For each query i ($1 \leq i \leq Q$), you need to determine the number of vertices V that satisfy the following conditions:

- Vertex U_i is present in the shortest path from V to the root, including the endpoints.
- The shortest path from V to the root contains exactly D_i edges.

In other words, you need to find the count of vertices that have U_i vertex in their shortest path to the root and contain exactly D_i edges from the root.



The example in the sample

If the query is $U_i = 1, D_i = 2$, so we need to count the number of vertices V that vertex U_i appears in the shortest path from V to the root and the edges between the root and vertex V is exactly D_i, so the answer will be 3, and they are $[4, 5, 7]$.

Input

In the first line of input you will be given N ($1 \leq N \leq 2 \cdot 10^5$) number of vertices

Then there are $N - 1$ lines describing the tree.

In the line i there are two integers U_i and V_i ($1 \leq U_i, V_i \leq N$) — edge in tree.

In the third line of input, you will be given Q $(1 \le Q \le 2 \cdot 10^5)$

The following Q lines you will be given:

- U_i $(1 \le U_i \le N)$ Vertex U_i is represented in the shortest path from V to the root, including the endpoints
- D_i $(0 \le D_i \le N - 1)$ The shortest path from V to the root contains exactly D_i edges

Output

For each query, print the number of vertices that have U_i vertex in their shortest path to the root and contains exactly D_i edges from the root.

Example

standard input	standard output
7 1 2 1 3 2 4 2 5 4 6 2 7 4 1 2 7 3 4 3 2 2	3 0 1 3

Problem P. N!

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Green`

In the realm of Skaadia, the legendary adventurer 7oSkaa faced a mathematical puzzle. They sought the meaning behind the expression $\frac{N!}{(N-2)!}$ hidden within a tale. With unwavering determination, 7oSkaa discovered its true essence: the number of ways to select two elements from a set of n. Their triumph left an indelible mark on the world of mathematics, inspiring others to explore the wonders of combinations and permutations.

$N!$ means factorial of N.

Input

The first line contains an integer T $(1 \le T \le 10^5)$ — number of test cases.

The next t lines, each line contains an integer N $(1 \le N \le 10^6)$.

Output

For each test, print one line contains the number of ways to select two elements from a set of N.

Example

standard input	standard output
3 4 7 10	12 42 90