A. LIS Count

CPU: 1s, Memory: 2048MB

Given a permutation, you can easily find its LIS (Longest Increasing Subsequence).

You can also find the number of subsequences which is a LIS of the given permutation.

But the life of a programmer is not so easy, huh!

So, given a number **C**, you have to find **a permutation** such that **the number of LIS in this permutation is C**.

Input

First line contains the number of test cases **T** **($1 \le T \le 5000$)**.

The following **T** lines each contains a single integer **C** **($1 \le C \le 10^{18}$)**.

Output

For each test case, find a permutation that has the desired LIS count and output two lines:

- First line should contain the size of the permutation **N** **($1 \le N \le 200$)**.
- Second line should contain N distinct integers **a_i** **($1 \le a_i \le N$)** denoting the permutation.

If there are many solutions, you can print any of them. If there is no such permutation, print "-1" in a single line (without the quotes).

Samples

Input	**Output**
2 2 1	7 4 2 7 1 3 5 6 4 1 2 3 4

B. Colorful Graph Counting

CPU: 1s, Memory: 2048MB

You are given three numbers **N**, **K** and **L**. You have to calculate the number of graphs such that -

- The graph is an undirected, simple graph (no multi-edge and no self-loops),
- The graph consists of exactly **N** nodes,
- Each node will have a color which is denoted by an integer in the range **[1, K]**,
- For any node **u** having color c_u and any other color **p** != c_u , there exists **at least L nodes** with color **p** reachable from node **u**.
- Node **a** is considered to be reachable from node **b**, if there exists a sequence of edges which connects node **a** and node **b**.

For example, when **K = 3** and **L = 5**, this means:

- For any node of color 1, there must be at least 5 nodes of color 2 and at least 5 nodes of color 3 reachable from that node.
- For any node of color 2, there must be at least 5 nodes of color 1 and at least 5 nodes of color 3 reachable from that node.
- For any node of color 3, there must be at least 5 nodes of color 1 and at least 5 nodes of color 2 reachable from that node.

Two graphs are considered to be different if either of the following holds true -

- There is at least one node such that color of that node is different in two graphs,
- There is at least a pair of node **(u,v)** such that there is an edge between **u** and **v** in one graph and not in another graph.

As the result can be very large, output it modulo **998244353**.

Input

The only line of input contains **3** integers **N**, **K** and **L** (**1 ≤ N, K, L ≤ 5000**).

Output

Output the number of graphs modulo **998244353**.

Samples

Input	Output
4 2 2	228

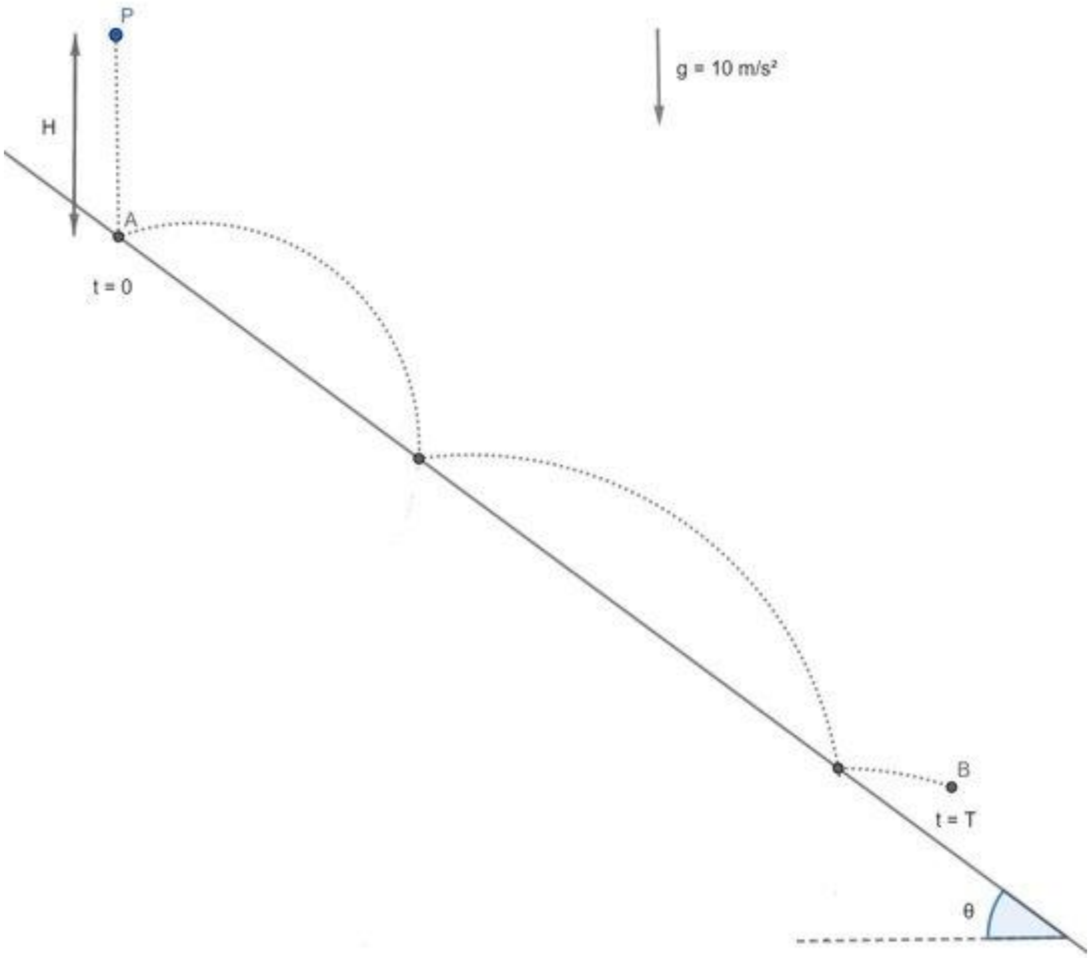
Input	Output
9 2 2	559252829

C. Rock and Roll

CPU: 1s, Memory: 2048MB

A ball is dropped from point **P** on an inclined surface with an inclination angle of **θ degrees**. At first the ball hits the surface on point **A** from height **H meter** at **time = 0**. At **time = T seconds** the ball is at point **B**. You have to find the distance of point **B** from point **A** in **meter**.

You may assume that the gravitational acceleration **($g = 10\ ms^{-2}$)** is constant everywhere, all collisions are perfectly elastic, air resistance is negligible, the inclined surface is infinitely large and the ball is a point object.



Note that the image is not drawn to the scale but only for illustration purpose.

Input

The first line contains one integer N denoting the number of test cases.

Each of the next N lines contain three integers denoting H, T, θ respectively.

Constraints

$1 \le N \le 10^5$

$1 \le H \le 100$

$1 \le T \le 100$

$0 < \theta < 90$

Output

For each test case, print one line containing a real number denoting the distance from point A to point B.

Your answer is considered correct if its absolute or relative error doesn't exceed 10-6.

Sample

Input	Output
4 1 1 45 1 2 45 1 3 45 1 10 45	6.704280537837 20.473046133885 41.311703351602 385.176394103919

D. A Splash of Rain

CPU: 1s, Memory: 2048MB

You have an array **A** (1-indexed) of **N** integers. You can perform any number of operations on this array. An operation goes like this,

(1) You pick an index **i** such that **1 ≤ i ≤ n-2** and **a[i] + a[i+2] > a[i+1]**

(2) Then you swap **a[i]** and **a[i+2]**

You have to output if you can sort the array in non-decreasing order after performing any number of operations.

Input

The first line contains one integer **T** denoting the number of test cases.

For each test, the first line contains **N**, the size of the array. The next line contains **N** integers denoting the elements of the array.

Constraints

$1 \le T \le 10^5$

$1 \le N \le 10^5$

$1 \le$ Sum of N over all testcases $\le 10^5$

$1 \le A_i \le 10^9$

Output

For each test case, print **"YES"** (without quotes) if it is possible to sort the array in non-decreasing order using any number of operations. Otherwise print **"NO"** (without quotes).

Sample

Input	**Output**
3 2 12 10 4 4 2 3 1 4 3 4 1 2	NO NO YES

E. Shelter Home

CPU: 4s, Memory: 1024MB

Batman protects the people of Gotham from various superhuman villains. Apart from superhuman villains, another problem arose in Gotham. Too many people lost their homes because of the fight between batman and superhuman villains, they fought and demolished buildings after buildings. But batman is very wealthy and generous too. So he made a list of **n** people whose house he broke. He wants to give them a new home. He also wants to minimize the total number of homes built. So he gave the optimal home assignment task to Robin, his son. Robin numbered the **n** people from **0** to **n-1**. Then he gave batman **m** pair of number (i, j). where **0 ≤ i, j ≤ n-1** and **i ≠ j** . Such a pair means i^{th} person wants to stay with j^{th} person in the same house and vice versa. Rest of the pairs don't like each other so they won't stay together. Now the next phase of assigning houses to people is given to you. You don't need to fulfill each person's wish of who they want to stay together but you also can't put two person in the same house if they don't want to stay together. You have to output the minimum number of house needed to build so every person gets a house. Assume each house can hold maximum of **k** people.

Input

First line will contain the number of test case **t (0 < t ≤ 10).** First line of each test case contains three numbers **n (0 ≤ n ≤ 20)** , **m (0 ≤ m ≤ n * (n-1)/2)** and **k (0 < k ≤ 20)** defined as above, then the following m lines each will contain two integer **i** and **j**. That means i^{th} person and j^{th} person wants to stay together. Here **0 ≤ i, j < n** and **i ≠ j**.

Output

For each test case output only one line, the minimum number of houses need to be built.

Sample

Input	Output
2 2 1 1 1 0 5 3 3 0 1 1 2 3 4	2 3

F. Loba's Revenge

CPU: 2s, Memory: 2048MB

Revenant, a robot assassin, killed Loba's parents when she was just a child, and since then she has been meticulously hunting down Revenant in underground bunkers of Olympus in a hate-fueled quest for revenge. Revenant has lived for thousands of years, killing hundreds of people. Now, he wants to just die. But how can a robot die? The secret lies in his source code, which is hidden from him by his creator. Loba, with her revengeful mind, found the source code on the World's Edge. For Revenant, this means he will be damned to eternal life, as his source code was the only way to end his existence. So, Loba threw it through a phase runner in King's Canyon, making it pretty much untraceable. Infuriated, Revenant begins to attack Loba. So, she hides in the secret toxic chamber of Caustic, creator of King's Canyon and a friend of Loba.

Eventually, Revenant found the chamber of Caustic. Since Caustic enjoys nothing except suffering of others, he has given Revenant the impossible task of passing through the phase runner. For that, he has also given Revenant the map of the phase runner. The phase runner consists of **n** one dimensional chambers, numbered from **1** to **n**, where the $\mathbf{i^{th}}$ and $\mathbf{(i+1)^{th}}$ chambers are next to each other. Each chamber is toxic and the toxicity of $\mathbf{i^{th}}$ chamber is given by an integer $\mathbf{A_i}$. Revenant knows the toxicity of each chamber. A Normal person cannot survive these toxic chambers but Revenant is a robot, no problem for him. But even for him, it is impossible to move through these toxic chambers without the help of Watson, the daughter of Caustic, as she put electric fences everywhere in the chambers. Due to the nature of gas, these electric fences require special type of batteries.

She has limited amount of batteries. She needs to calculate the number of batteries needed to power the fences so that she can optimally use the batteries. If the power of a battery type is **g**, the number of batteries needed to power the fences from $\mathbf{L^{th}}$ chamber to $\mathbf{R^{th}}$ chamber is the length of longest sub segment **A[i], A[i+1]....A[j]** such that **L ≤ i ≤ j ≤ R** and **gcd(A[i], A[i+1]....A[j]) = g**. Let's define this length as **C** i.e. **j - i + 1 = C**. If there exists no such sub segment, then **C = 0**.

She has given Revenant **Q** queries. Upon answering these queries, she will gift him the portal of Wraith. Portal of Wraith is a tunnel in fourth dimensional space, which allows the traveller to pass through the fences and toxic gases without being harmed.

Input

First line contains an integer **n**, number of elements of array **A**.

Then the second line contains n space separated integers that represent **A[1], A[2] …… A[n]**

Third line contains an integer **Q**, the number of queries. Each of the next **Q** line contains three integers **g, L, R**.

Output

For each query, we get a value of **C**. Your task is to print only one line, **XOR** of all C's.

Constraints

$1 \le n \le 50000$

$1 \le A[i] \le 10^{18}$

$1 \le Q \le 100000$

Sample

Input	Output
6 3 3 3 6 12 3 9 3 1 3 3 2 5 3 3 6 6 4 6 6 5 6 3 4 6 3 1 6 5 5 5 12 5 5	5

Note: Greatest common divisor (gcd) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers and gcd(a, b, c) = gcd(gcd(a, b), c).

G. K-ary Game

CPU: 1s, Memory: 2048MB

Alice and Bob will play a game on a **perfect K-ary rooted tree** of **depth D**, where **K is even**. This means -

- Each non-leaf node has exactly **K** children.
- Each leaf will be at the same depth **D**.
- Depth of root is **0**.

Initially both of them have 0 points. In each move, a player can delete exactly one node and this node's degree will be added to his total point. Deleting a node will automatically remove its' adjacent edges. Each node can be deleted at most once.

Each player wants to maximize his own points. Both players will play optimally.

If Alice moves first, output the maximum point that Alice can achieve.

Input

The first line will contain a single integer **T ($1 \le T \le 10^5$)**.

Each test case will contain two integers denoting **K ($2 \le K \le 10^9$, K is even)** and **D ($1 \le D \le 10^9$)**.

Output

For each test case, output in a single line the maximum point Alice can achieve.

Since the answer can be pretty large, output the value modulo **10^9**.

Sample

Input	Output
5 4 1 2 1 4 2 2 4 2 3	4 2 12 16 8

H. At Least Once

CPU: 1s, Memory: 2048MB

Given a string **S** consisting of lowercase english letters, find a subsequence of **S** which contains each letter present in **S** at least once. If there are multiple such strings, output the lexicographically minimum one.

Input

The first line will contain a single integer **T** (**1 ≤ T ≤ 10**).

Each of the following **T** lines each contains a single string **S** consisting of lowercase English letters (**1 <= |S| <= 10^4**)

Output

For each test case, print in a single line the lexicographically minimum subsequence of **S** which contains each letter present in **S** at least once.

Samples

Input	Output
3 abcd abca acbbc	abcd abc abbc

Input	Output
7 aaabbbcccddd dbca acbdab dbcddaabc acbdacbd aabbcababdabb bssaaabbb	aaabbbcccd dbca acbd bcda aacbd aabbcaabd bsa

I. Caching Research

CPU: 2s, Memory: 1024MB

Ted, the great computer scientist was recently doing some serious research on caching mechanism. He had some data, which can be represented as a permutation of size **n**, he decided to make some operations on it.

In each operation,

- He chooses some index i $(1 \le i \le n)$,
- Moves **P_i** to the front of the permutation.

For example, let's assume the initial permutation is **{4, 1, 5, 3, 2}** and the index sequence of operations is **(2, 5, 4)**.

So after the 1st operation, the permutation becomes **{1, 4, 5, 3, 2}**.

After the 2nd operation, the permutation becomes **{2, 1, 4, 5, 3}**.

After the 3rd operation, the permutation becomes **{5, 2, 1, 4, 3}**.

After making some operation on a permutation, he left his room for some time. When he got back he could see the initial permutation and also the final permutation but he couldn't remember which operations he made. But he remembers how many operations he made. Now he wonders, how many possible sequence of operations are there which would take him from the initial permutation to the final permutation. As the number can be quite large, output it modulo **998244353**.

Input

First line of input consists of a single integer **T** **($1 \le T \le 3$)**, the number of test cases.

Each test case consists of two lines as follows:

- First line of the test case contains **n**, the length of permutation and **m**, the length of the operation sequence **($1 \le n, m \le 10^5$)**.
- Second line contains **n** space-separated distinct integers, denoting the initial permutation **A** **($1 \le A_i \le n$)**.
- Third line contains **n** space-separated distinct integers, denoting the final permutation **B** **($1 \le B_i \le n$)**.

Output

For each testcase print one integer, the possible number of operation sequences of length **m** modulo **998244353**.

Two sequences are considered different if there is at least one index where they differ.

Sample

Input	Output
1 4 2 3 2 4 1 1 3 2 4	2

We define a sequence of integers of length **n** as a permutation **P** if every integer from **1** to **n** appears exactly once in the sequence.

J. Sorting Again

CPU: 1s, Memory: 1024MB

You are given an array **A** (1-indexed) of **n** non-negative integers.

In one operation, you pick two indices **i, j** such that **1 ≤ i, j ≤ n** and **i + j** is **odd**, then you can swap **Ai** and **Aj**.

You can make as many operations as you want.

You want to sort the array in **strictly increasing** order. Can you do it ?

Input

The first line contains one integer **T**, denoting the number of test cases.(**1 ≤ T ≤ 10**)

For each case, the first line contains an integer **n**, the size of the array (**1 ≤ n ≤ 50**).

The next line contains **n** integers denoting the elements of the array **(0 ≤ Ai ≤ 10^3)**.

Output

In a single line print **"YES"**, if it is possible to sort the array with the allowed operations.

Otherwise print **"NO"** (without the quotes).

Sample

Input	**Output**
2 4 8 3 7 2 4 2 3 5 2	YES NO

Note: In the first case, we can make a swap between **1st** index and **4th** index as **1 + 4 = 5** is an **odd** number.