

Problem A. GCD MEX

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

You are given a positive integer n . Your task is to construct an array a of size $|a|$ such that $|a| \leq n$.

Let b be an empty array. For every pair of indices i and j such that $1 \leq i < j \leq |a|$, we add the greatest common divisor (GCD) of a_i and a_j to array b .

Your constructed array a must satisfy the condition that the MEX of the array b (containing all pairwise GCDs $\gcd(a_i, a_j)$ for $i \neq j$) is at least n .

The MEX (Minimum Excluded value) of a set of positive integers is the smallest positive integer that is not present in the set. **The MEX of an empty set of positive integers is 1.**

Input

The input consists of t test cases. The first line contains a single integer t ($1 \leq t \leq 30$) – the number of test cases. Each of the following t lines contains a single integer n ($1 \leq n \leq 30$).

Output

For each test case, output two lines. The first line should contain a single integer, the size of the constructed array a . The second line should contain $|a|$ space-separated integers, the elements of the array a . The elements a_i must satisfy $1 \leq a_i \leq 10^{18}$.

Example

standard input	standard output
2	2
2	1 2
3	3
	1 2 4

Problem B. Random Shuffle?

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

We have an array a of n integers, and an initially empty sheet of paper.

We will do the following operations in-order m times:

- We shuffle the array a .
- We choose a random positive integer $k \leq n$.
- We choose a random index $1 \leq i \leq k$.
- We write a_i at the end of the paper.

You are asked to calculate the probability that the integers written on the sheet are non-decreasing. Output the calculated value modulo 998244353.

Input

The first and only line contains two positive integers n and m ($1 \leq n, m \leq 10^6$) — the length of the array and the number of times we repeat the operations.

The following line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$) — the elements of the array a .

Output

Output one integer, the probability that the integers written on the sheet are non-decreasing modulo 998244353.

Example

standard input	standard output
3 2 1 1 2	110916040

Note

The possible final sequences on the paper are :

- $(1, 1)$ with probability $\frac{4}{9}$
- $(1, 2)$ with probability $\frac{2}{9}$
- $(2, 1)$ with probability $\frac{2}{9}$
- $(2, 2)$ with probability $\frac{1}{9}$

Thus the answer is $\frac{4}{9} + \frac{2}{9} + \frac{1}{9} = \frac{7}{9}$.

Problem C. Pizza Man

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

There is a Pizza Man, who sells pizza, called Hosen. Hosen can store no more than m pieces of pizza and initially he has m pieces of pizza stored in his store.

There are n customers; the i_{th} of them will order a_i pieces of pizza.

And Hosen has a positive integer d , that is when the number of pieces in the store is less than d , he refills it to m .

Here is a more detailed explanation.

When a customer orders x pieces of pizza, the following happens:

- If Hosen has at least x pieces of pizza stored in the store, he sells x pieces to the customer and now the pieces of pizza Hosen has will decrease by x . In this case, the customer will be happy.
- If Hosen has less than x pieces of pizza stored in the store, he sells all the remaining pieces of pizza to the customer and now he has zero pieces of pizza. In this case, the customer will be sad.
- After serving each customer, if Hosen has less than d pieces in the store, he will refill the store to have m pieces of pizza.

You are given m, n and the array of customers a ; find the minimum value of d for which all the customers will be happy.

Input

The first line contains a single integer t ($1 \leq t \leq 10^5$), the number of test cases.

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 2 \times 10^5$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq m$), the elements of the array a .

Let s be the sum of the array elements. It is guaranteed that the sum of s overall test cases doesn't exceed 2×10^5 .

Output

For each test case output a single positive integer d , the minimum value of d for which all the customers will be happy.

Example

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

Problem D. Master of the Arena

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

You are the master of an arena with n fighters under your control, numbered from 1 to n . Fighter 1 is your favourite—you want them to emerge as the sole champion.

You are given an $n \times n$ matrix M . For $i \neq j$:

- $M_{i,j} = 1$ means fighter i definitely beats fighter j .
- $M_{i,j} = 0$ means fighter i definitely loses to fighter j .
- $M_{i,j} = ?$ means the outcome is undecided and you may choose it.

It is guaranteed that $M_{i,i} = 0$ for all i , and for all $i \neq j$, the pair $(M_{i,j}, M_{j,i})$ is one of: $(1, 0)$, $(0, 1)$, or $(?, ?)$.

You must schedule exactly $n - 1$ one-on-one matches so that in the end exactly one fighter remains undefeated, and that fighter must be number 1. A fighter may appear in multiple matches (until eliminated).

Decide any choices for the '?' entries and output a sequence of $n - 1$ matches (a, b) meaning " a defeats b ". If it's impossible to make fighter 1 the unique champion, print **No**.

Input

The first line contains a single integer t ($1 \leq t \leq 500$) — the number of test cases.

The first line of each test case contains integer n ($2 \leq n \leq 1000$) — the number of fighters.

Then in each test case follow n lines each with a string of length n , the i -th line being $M_{i,1}M_{i,2} \dots M_{i,n}$.

It is guaranteed that the sum of n over all test cases doesn't exceed 1000.

Output

In each test case, if it is impossible to guarantee fighter 1 wins, print **No**. Otherwise, print **Yes** followed by $n - 1$ lines. Each line should contain two integers a and b indicating that fighter a defeats fighter b in a scheduled match. You may print each character in either case, for example **YES** and **yEs** will also be accepted.

Example

standard input	standard output
3	Yes
3	2 3
010	1 2
001	
100	Yes
5	4 5
01000	3 4
00100	2 3
100?0	1 2
11?01	
11100	No
5	
01000	
00100	
10000	
11101	
11100	

Problem E. Clean White Paths

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

Given a tree of n vertices, each node is initially either black or white.

A path from node u to node v is:

- white if both nodes u and v are white.
- black if both nodes u and v are black.
- otherwise it is a normal path neither white nor black.

A white path is clean if and only if there is no black path that contains this white path as a subpath of it.

A path is a subpath from another path if the set of vertices of this path forms a subset of the vertices of the other path.

The score of the tree is the number of Clean White Paths.

Note that path (u, v) is the same as path (v, u) and has to be counted once, and you have to count paths of the form (u, u) .

Given q update queries, in each query you are given an integer u , make the node u black.

After each query, print the score of the tree.

Input

The first line contains a single integer t ($1 \leq t \leq 10^5$), the number of test cases.

The first line of each test case contains two integers n and q ($1 \leq n \leq 3 \times 10^5$) ($1 \leq q \leq 3 \times 10^5$), the size of the tree and the number of queries respectively.

The second line of each test case contains a binary string s of size n ($s_i \in (0, 1)$), if s_i is 1 then the i_{th} node is white, otherwise the node is black.

The next $n - 1$ lines describe the edges of the tree, each line contains two integers u, v ($1 \leq u, v \leq n$), which means there is an edge between node u and node v .

Each of the next q lines contains a single integer u ($1 \leq u \leq n$), the node to make black.

It is guaranteed that the sum of n and the sum of q over all test cases doesn't exceed 3×10^5 .

Output

In each test case, after each query print the score of the tree.

Example

standard input	standard output
1	10
5 5	6
11111	2
5 2	2
1 3	0
5 4	
3 4	
4	
3	
2	
4	
1	

Note

After the first update, node 4 is black; all white paths are clean, so the answer is 10.

After the third update, there are two clean white paths: (1,5) and (1,1).

Problem F. Meen 3mk?

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

In many parts of the world, when someone absolutely dominates a field—be it sports, coding, cooking, or making memes—we say they're **your uncle**.

It's not a family thing. It's a **respect thing**. Like, "*Tourist is your uncle*" means he owns Competitive Programming, and you just watch and clap.

So today, we're not solving graphs or DP. We're solving **respect**.

Given a simple question, answer it from your heart.

Input

Who is your uncle?

Output

Print the name of your uncle, or just print "Tourist".

Example

standard input	standard output
Who is your uncle?	Tourist

Problem G. Nim Game In Byteland

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

There are n cities, numbered from 1 to n . City 1 is where *Alice* lives, and city n is where her rival *Bob* resides.

The cities are connected by directed roads. Each city has **at most two** outgoing roads.

Alice wants to move from city 1 to city n . She can visit the same city multiple times. In each city she visits, she can choose to buy the pile in that city, but she can buy **at most one pile per city** during her journey.

When Alice reaches city n , she will play a **Nim game** against Bob using the piles she has collected. **Bob plays first**.

Your task is to determine whether Alice can select piles along her journey such that she **guarantees a victory** over Bob.

Note: Alice must select at least one pile on the path from 1 to n , and it's guaranteed there's a path from 1 to n .

In Nim, players take turns removing any number of stones from one pile. The player who removes the last stone wins.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the number of cities.

Each of the next n lines contains two integers u_i and v_i ($0 \leq u_i, v_i \leq n$) — the outgoing roads from city i .

- If a city has one outgoing road, the second number will be 0.
- If a city has no outgoing roads, both numbers will be 0.

The next line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^6$) — the number of stones in the pile at each city.

Output

output YES if Alice can guarantee a win, and NO otherwise.

Examples

standard input	standard output
4 2 0 3 4 1 0 0 0 2 4 1 1	YES
4 2 0 3 4 1 0 0 0 2 4 8 1	NO

Problem H. Minimum Path

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

Given an array a of length n ($1 \leq n \leq 10^5$) ($1 \leq a_i \leq 10^9$). Initially, you are at index 1 and you want to go to index n and visit every index **exactly once**.

You can jump from your current index i to another index j if and only if the absolute difference doesn't exceed 2, that is $|i - j| \leq 2$.

You also have an array b , that initially contains one element a_1 . When you jump to a new index j , you mark it as visited and put a_j at the end of the array b .

Your task is to print the lexicographically minimum array b that can be obtained after a valid walk, that is after visiting all indices and **ending at index n** .

Input

The first line contains a single integer t ($1 \leq t \leq 10^5$), the number of test cases.

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

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

It is guaranteed that the sum of n over all test cases doesn't exceed 10^5 .

Output

For each test case, print n space separated integers $b_1, b_2, \dots, b_n$, the elements of array b , that is the lexicographically minimum array that can be obtained after a valid walk.

Example

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

Problem I. Reverse and Remove

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

You are given an array a of length n and a positive integer k .

You will perform the following operation on the array exactly k times:

- Remove the first element of the array.
- Reverse the remaining array.

After performing the operation exactly k times, print the resulting array.

Input

The first line contains two integers n and k ($1 \leq k < n \leq 10^5$) — the size of the array and the number of operations.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — the elements of the array.

Output

Print $n - k$ integers — the elements of the resulting array after performing the operations.

Example

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

Problem J. Prefix GCD

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

For an array a of length n , we define the function $f(a)$ as following:

$$f(a) = gcd(a_1, a_1) + gcd(a_1, a_2) + ..gcd(a_1, a_2, ..a_n)$$

You are given two positive integers n and m . You need to output the sum of $f(a)$ over all arrays a satisfying the following:

- The length of the array a is equal to n .
- $1 \leq a_i \leq m$ for each $1 \leq i \leq n$.

For two positive integers x and y , $gcd(x, y)$ equals the greatest integer that divides both x and y .

Input

The first line contains a single positive integer number t ($1 \leq t \leq 10^3$) — the number of testcases.

The first line of each testcase contains two positive integers n and m ($1 \leq n, m \leq 10^6$).

Output

For each testcase, output in a separate line the answer described in the statement. Since the output number can be too large, output it modulo $10^9 + 7$.

Example

standard input	standard output
2	15
1 5	550
2 9	

Problem K. And X Elements

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

You are given an array a of n elements and an integer x . You also have a variable v which is initially equal to 0.

You will perform exactly one operation for each element of the array, and the operations must be applied **in the order of the array** (i.e., from a_1 to a_n). For each i from 1 to n , you can choose one of the following operations:

1. $v = v \mid a_i$ (bitwise OR)
2. $v = v \& a_i$ (bitwise AND)

You must use the **second operation (AND)** at least x times throughout the process.

Determine the **maximum possible value of v** at the end.

Input

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

The first line of each test case contains two integers n and x ($1 \leq x \leq n \leq 10^5$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^5$), the elements of the array.

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

Output

For each test case, print in a separate line the maximum value possible for v after the n operations.

Example

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

Problem L. Equalize

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

You are given an array a of n non-negative integers, and a positive integer m .

You need to answer q queries. In each query, you are given a non-negative integer x .

You can do the following operation:

- choose two integers $1 \leq l \leq r \leq n$ such that $r - l + 1 = m$, and for each $l \leq i \leq r$ replace a_i with $a_i | x$, where $x | y$ denotes the bit-wise OR of x and y .

You need to make all the elements of the array equal.

For each query, output the minimum number of operations needed to make the elements of the array equal, or output -1 if they cannot be made equal.

Input

The first line contains a single positive integer $1 \leq t \leq 3 \times 10^5$ — the number of testcases.

The first line of each testcase contains two positive integers n and m ($1 \leq m \leq n \leq 10^6$).

The next line contains n non-negative integers $a_1, a_2, \dots, a_n$ — the elements of array a , where ($0 \leq a_i < 2^{30}$).

The next line contains a single positive integer q ($1 \leq q \leq 10^6$) — the number of queries.

Each of the following q lines contains a single non-negative integer ($0 \leq x < 2^{30}$).

It is guaranteed that the sum of n and q over all testcases does not exceed 10^6 .

Output

For each query of each testcase, output the answer to it as described in the problem statement.

Example

standard input	standard output
2	1
2 1	-1
1 3	3
2	3
2	3
4	3
6 2	
22 4 14 30 28 12	
4	
26	
27	
30	
31	

Problem M. Maximum Or Permutation

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

You are given an integer n . Your task is to construct a permutation p of the set $\{1, 2, \dots, n\}$ such that when the permutation is considered as circular (i.e. p_1 and p_n are adjacent), the maximum bitwise OR over all adjacent pairs is minimized.

Formally, for a permutation $p = [p_1, p_2, \dots, p_n]$, define

$$f(p) = \max_{1 \leq i \leq n} (p_i \mid p_{i+1}),$$

where we define $p_{n+1} = p_1$ and $\mid$ denotes the bitwise OR operation. Your goal is to construct a permutation p such that $f(p)$ is as small as possible.

If there are multiple answers, you may output any one of them.

Input

The first line contains a single integer t ($1 \leq t \leq 10^5$) — the number of test cases.

Each of the following t lines contains a single integer n ($2 \leq n \leq 5 \cdot 10^5$) — the size of the permutation.

It is guaranteed that the sum of all n over the test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, output a permutation of $\{1, 2, \dots, n\}$ (i.e. a sequence of n distinct integers) such that the maximum bitwise OR of any two adjacent elements (considering the permutation circularly) is minimized.

Example

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