ACPC
AFRICA & ARAB
Collegiate Programming
Championship

International Collegiate Programming Contest
The 2023 Tunisian Collegiate Programming Contest
Sousse
June 2023

جامعة الدول العربية
الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري

The International Collegiate Programming Contest
Sponsored by ICPC Foundation

TCPC

The 2023 Tunisian Collegiate Programming Contest

(Contest Problems)

SOUSSE
Poly
technique
EUR-ACE®

Sousse
June 2023

Problem A. Fibonacci Sequence

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

We call a sequence A of lowercase Latin letters of length N a Fibonacci sequence if it can be written according to the following recurrence:

- $F_0 = S_1$.
- $F_1 = S_2$.
- $F_i = F_{i-2} + F_{i-1}$.
- $A = F_K = F_{K-2} + F_{K-1}$.

You are given a Fibonacci Sequence A, and you need to find both S_1 and S_2.

Note that in this problem:

- S_1 and S_2 are both of the same length.
- If there are no such valid S_1 and S_2 to make A a Fibonacci Sequence, print -1.
- If there are multiple possible answers, print the one with the shortest possible length of S_1 and S_2.
- $+$ refers to the concatenation operation. For example, the concatenation of sequences "ab" and "cd" is "ab" + "cd" = "abcd".

Input

The first line of input contains a single integer T $(1 \le T \le 10^4)$ — the number of testcases. Each testcase is described using two lines.

The first line of each testcase contains a single integer N $(1 \le N \le 5 \cdot 10^5)$ — the length of sequence A.

The second line of each testcase contains N lowercase latin letters — the sequence A.

It is guaranteed that the sum of N over all testcases doesn't exceed $5 \cdot 10^5$.

Output

For each testcase if no valid S_1 and S_2 exist, print -1 on a single line, otherwise print three lines.

The first line should contain an integer L representing the length of S_1 and S_2.

The second line should contain sequence S_1.

The third line should contain sequence S_2.

If multiple S_1 and S_2 exist, you must print S_1 and S_2 with the smallest possible L.

Example

standard input	standard output
3 5 abbab 5 abcac 4 acpc	1 a b -1 2 ac pc

Problem B. Bitwise Beautiful Arrays

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

We calculate the Bitwise Beauty of an array A as follows:

- Write down each possible **non-empty** subset of elements of A.
- For each possible subset, calculate the **bitwise OR** value of all its elements, call it X.
- The **Bitwise Beauty** is equal to the number of **distinct** possible values X that result from the previous calculation.

You are given an array A of length N and Q queries.

For the i_{th} query you are given an integer Y_i and then you do the following:

- Create a new **empty** array B_i.
- For each element of A that is a **submask** of Y_i, add it to array B_i.

The answer of i_{th} query is the **Bitwise Beauty** of array B_i.

The **bitwise OR** is a binary operation that takes two bit patterns of equal length and performs the **logical OR** operation on each pair of corresponding bits. The result in each position is 0 if both bits are 0, and 1 otherwise. For example: 0101 (decimal 5) OR 0011 (decimal 3) = 0111 (decimal 7).

Mask A is called a **submask** of a given mask B if all **set** bits of A are a subset of the **set** bits of B. e.g., 1010, 1000, 0010 are submasks of 1011, while 1100 is not.

A **set** X is a **subset** of a **set** Y if X can be obtained from Y by the deletion of several (possibly, zero or all) elements.

Input

The first line of input contains a single integer N $(1 \le N \le 10^6)$ — the length of array A.

The second line of input contains N space-separated integers $A_1, A_2, \ldots, A_N$ $(1 \le A_i \le 10^6)$ — the array A.

The third line of input contains a single integer Q $(1 \le Q \le 10^6)$ — the number of queries.

The fourth line of input contains Q space-separated integers $Y_1, Y_2, \ldots, Y_Q$ $(1 \le Y_i \le 10^6)$ — the given values in each query.

Output

For the i_{th} query print a single integer representing the **Bitwise Beauty** of array B_i.

Example

`standard input`	`standard output`

`6`
`3 1 4 1 5 9`
`10`
`1 2 3 4 5 6 7 8 9 10`
	`1 0 2 1 3 1 5 0 2 0`

Problem C. Divisibility

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

Coach Khaled is known in Syria for his solid skills regarding divisibility problems, so we asked him to prepare a beautiful, clear problem for you to solve. Here is what he prepared:

We call a pair of elements (X, Y) **Good Pair** if: $\{X > Y\}$ and $(X + Y)$ is divisible by $(X - Y)$.

You are given an array A of N **distinct** elements. Count the number of **Good Pairs**.

Input

The first line of input contains a single integer N $(1 \le N \le 10^6)$ — the length of array A.

The second line of input contains N space-separated **distinct** integers $A_1, A_2, \ldots, A_N$ $(1 \le A_i \le 10^6)$ — the values in array A.

Output

Print a single integer — the number of **Good Pairs** in array A.

Examples

standard input	standard output
3 5 1 7	1
4 6 2 5 8	3

Problem D. Game of Permutations

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

Obada and Osama are playing the *Game of Permutations*. At the beginning of the game, they have:

- A Permutation P of length N.
- An integer S that equals 0.

During each turn, the player in turn has to:

- Pick some index i $(1 \leq i \leq N)$ such that $i \neq P_i$.
- Either add P_i to S, or subtract P_i from S.
- Swap values at indices (i, P_i), i.e swap (P_i, P_{P_i}).

The game ends when P is sorted in increasing order. Obada plays first, and he wants to maximize the value of S. On the other hand, Osama wants to minimize the value of S. Both players play optimally.

We define the **Score** of the game that starts with permutation P as the value S after the game ends.

Obada and Osama have been playing this game all week. They played starting with every possible permutation P of length N. What is the sum of **Scores** over all the games they played?

The answer might be very large, so you need to print it modulo $10^9 + 7$.

Input

The first line of input contains a single integer T $(1 \leq T \leq 10^6)$ — the number of testcases. Each testcase is described using a single line.

The only line of input in each testcase contains a single integer N $(1 \leq N \leq 10^6)$ — the length of permutation P.

Note that there is no constraint on the sum of N over all testcases.

Output

For each testcase print a single integer — the sum of **Scores** over all possible permutations P modulo $10^9 + 7$.

Example

standard input	standard output
4 1 2 5 10	0 2 324 18950400

Problem E. Ultra Palindrome Sequence

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

We define a sequence S of lowercase Latin letters to be a **Palindrome** if it reads the same backward as it does forwards. For example, "abcba" is a **Palindrome**, while "acd" and "cca" are not.

We define a sequence S of lowercase Latin letters to be a **Super Palindrome** if it becomes **Palindrome** after **exactly** one shift to the left, i.e after taking the first character and appending it at the end of the sequence. For example, "aabcb" is **Super Palindrome** because it equals "abcba" **after** applying **one shift** to the left, and "abcba" itself is a **Palindrome**.

We define a sequence S of lowercase Latin letters to be **Ultra Palindrome** if it's either **Palindrome** or **Super Palindrome**.

You will be given a sequence S that consists of lowercase Latin letters and some **missing** letters represented as "?". You need to count the number of possible ways to replace **all** missing letters with lowercase Latin letters such that the resulting sequence is **Ultra Palindrome**.

Find the answer and print it modulo $10^9 + 7$.

Input

The first line of input contains a single integer T $(1 \le T \le 10^4)$ — the number of testcases. Each testcase is described using two lines.

The first line of each testcase contains a single integer N $(1 \le N \le 10^5)$ — the length of sequence S.

The second line of each testcase contains the sequence S.

It is guaranteed that the sum of N over all testcases doesn't exceed 10^5.

Output

For each testcase print a single integer — the number of possible ways to obtain an **Ultra Palindrome** sequence modulo $10^9 + 7$.

Example

standard input	standard output
3 5 r?da? 3 ?a? 4 a??b	2 51 1

Problem F. Weird Koko

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

Koko likes building arrays when he feels bored. What makes Koko annoyed is that each time he builds an array, he wants to ask some weird question about this array to his friend Wael.

This time Koko built his array A of length N in the following way:

- For each index i ($1 \leq i \leq N$) he chooses a **random value** between 1 and M as its value A_i. All values $\{1, 2, \ldots, M\}$ are **equiprobable**.

As usual, he came to Wael and asked him some weird question: Can you find me the **expected** number of **subsets** of A such that:

- The **sum** of its elements is **divisible** by M.
- The **sum** of its elements is **not equal** to M.

Unfortunately, Wael could not respond to Koko's question this time as he is very busy. You need to find the answer for Koko so that he does not get mad at his friend Wael.

A **set** X is a **subset** of a **set** Y if X can be obtained from Y by the deletion of several (possibly, zero or all) elements.

Input

The first line of input contains a single integer T ($1 \leq T \leq 10^5$) — the number of testcases. Each testcase is described using a single line.

The only line of each testcase contains two integers N and M ($1 \leq N, M \leq 10^5$) — the length of the array A and the number M.

It is guaranteed that the sum of N and the sum of M over all testcase doesn't exceed 10^5.

Output

For each testcase print a single integer — the expected number of the subsets that satisfy the conditions modulo $10^9 + 7$.

Formally, let $S = 10^9 + 7$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \bmod S$. Print the integer equal to $p \cdot q^{-1} \bmod S$. In other words, print an integer x such that $0 \leq x < S$ and $x \cdot q \equiv p \bmod S$.

Example

standard input	standard output

5
1 1
2 2
2 1
3 3
100 100
	1
	250000003
	2
	296296300
	784727514

Note

for the first testcase, there's only one array possible [1], and only one subset from this array satisfies the conditions which is the empty array ϕ

for the second testcase, there are four possible arrays:

- $[1, 1]$ which has only one subset: ϕ
- $[1, 2]$ which has only one subset: ϕ
- $[2, 1]$ which has only one subset: ϕ
- $[2, 2]$ which has two subsets: ϕ and $[2, 2]$

so in total the expected number of subsets is $\frac{5}{4}$

Problem G. Help The Judges

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

Choosing problems for a contest is no simple task, especially if you are preparing more than one contest at the same time!

The judges of the TunisianCPC are also preparing another contest along with this one. They have N problems to choose from, the i_{th} problem has a beauty equal to A_i.

Now as you might have already guessed, the judges want to divide the N problems into exactly two **non-empty** problemsets, such that every problem belongs to exactly one problemset.

The judges believe that the **quality** of **one problemset** is equal to the **difference** between the beauties of the problem with **maximum** beauty and the problem with **minimum** beauty in that problemset.

They also believe that quality is more important than quantity, so the two problemsets **don't have to be of the same size, but they must have equal qualities.**

The judges are busy working on the problems, so they asked you to write a program to check if the problems can be divided into two **(not necessarily of the same size)** problemsets with **equal qualities**.

Input

The first line of the input contains a single integer T $(1 \leq T \leq 10^5)$ — the number of testcases. Each testcase contains two lines.

The first line of each testcase contains a single integer N $(1 \leq N \leq 10^6)$ — the number of problems.

The second line contains N space-separated integers A_i $(1 \leq A_i \leq 10^6)$ — the beauty of the i_{th} problem.

It's guaranteed that the sum of N over all testcases doesn't exceed 10^6.

Output

For each testcase, print "YES" (without quotes) if the problems can be divided into two **non-empty** problemsets with equal qualities, otherwise print "NO" (without quotes).

You can print the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

Example

standard input	standard output
5	YES
5	NO
5 3 1 5 7	NO
1	YES
3	NO
4	
8 2 1 4	
2	
1 2	
3	
1 3 4	

Problem H. Palindrome Builder

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

You are given a sequence S of length N that consists **only of** the letters **'A'**, **'B'**, **and 'C'**. You can apply the following operation **at most once** for **each** index i $(1 \leq i \leq N)$ of the sequence S:

- If the character S_i is equal to **'A'**, change it to **'B'**.
- If the character S_i is equal to **'C'**, change it to **'B'**.
- If the character S_i is equal to **'B'**, change it to **either 'A'** or **'C'**.

Find the **lexicographically minimum palindrome** that you can get using the described operation **at each index at most once**.

We define a sequence S of lowercase Latin letters to be a **Palindrome** if it reads the same backward as it does forward. For example, "abcba" is a Palindrome, while "acd" and "cca" are not.

A sequence A is lexicographically smaller than a sequence B if and only if one of the following holds:

- A is a prefix of B, but $A \neq B$.
- In the leftmost position where A and B differ, the sequence A has a letter that appears earlier in the alphabet than the corresponding letter in B.

Input

The first line of input contains a single integer T $(1 \leq T \leq 10^4)$ — the number of testcases. Each testcase is described using two lines.

The first line of each testcase contains a single integer N $(1 \leq N \leq 10^5)$ — the length of sequence S.

The second line of each testcase contains a sequence S where (S_i is one of the letters **'A'**, **'B' or 'C'**).

It is guaranteed that the sum of N over all testcases doesn't exceed 10^5.

Output

For each testcase, print a single line containing a sequence P, the **lexicographically minimum palindrome** you can get with the provided operations.

Example

standard input	standard output
2 4 ABAB 5 ABCAB	AAAA AABAA

Problem I. Osama and the tree LIS

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

One day Osama was following the well-known technique to create problems that he learnt from his friend Tareq, ***aka "the blender of standards"****, when he came up with this problem.*

You will be given a tree of N vertices, each vertex has some value A_i written on it. The tree is rooted at vertex 1.

We define the beauty of some vertex U as the size of the longest sequence of vertices S satisfying all the following conditions:

- U is part of S.
- U is an ancestor of each vertex in S. (note that a vertex is an ancestor of itself).
- For each pair of different vertices (V_1, V_2) in S, either V_1 is an ancestor of V_2, or V_2 is an ancestor of V_1.
- If we sort S in increasing order of the depth of each vertex, and write down all the values written on the vertices in the order they appear after S is sorted, we get a **strictly increasing sequence**. (a sequence where each element is strictly greater than the element before it).

Your task to print the beauty of each vertex from 1 to N.

Note that the input is given in a modified way.

Input

The first line of the input contains a single integer N $(1 \le N \le 10^5)$ — The number of vertices in the tree.

The next line contains N space-separated integers $B_1, B_2, \ldots, B_n$ $(0 \le B_i \le 10^9 + 6)$ (the array B, not A). If i is a leaf, then $A_i = B_i$. Otherwise, $A_i = (B_i \cdot sum_i) \bmod 10^9 + 7$. Where sum_i is the sum of beauty of all the vertices in the subtree of i (except for vertex i).

The next $N - 1$ lines contain the edges of the tree. Each line contains two integers U and V denoting an edge between vertices U and V $(1 \le U, V \le N, U \ne V)$. It is guaranteed that these edges form a tree.

Output

Print a single line containing N space-separated integers — $beauty_1, beauty_2, \ldots, beauty_N$.

Example

standard input	standard output

3
1 2 7
1 2
2 3
	2 2 1

Problem J. Is It Helal Sequence?

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

We define a sequence of elements to be a **Helal Sequence** if the multiplication of **all** its elements is **divisible** by X.

You will be given an array A of length N and an integer X. Is there any **non-empty** subset of A that can be called a **Helal Sequence**?

Input

The first line of input contains a single integer T ($1 \leq T \leq 1000$) — the number of testcases. Each testcase is described using two lines.

The first line of each testcase contains two integers N and X ($1 \leq N \leq 10^5; 1 \leq X \leq 10^9$) — the length of the sequence A and the value X.

The second line of each testcase contains N space-separated integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 10^9$) — the array A.

It is guaranteed that the sum of N over all testcases doesn't exceed 10^5.

Output

For each testcase print "YES" if the seqeuence A contains a subset that is a **Helal Sequence** otherwise print "NO".

You can print the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

Example

standard input	standard output

3
3 4
5 4 10
3 12
10 21 10
3 7
3 6 4
	YES
	YES
	NO

Problem K. Btunis Beek

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Golden (Mustard)`

Wael was more than happy to hear that he is preparing problems for Tunisia. He instantly remembered the song of his childhood, and that magical line which says: “BTUNIS BEEK, WE ENTA M3AYA”.

However, as you know, too much happiness leads to forgetting some stuff. Wael forgot the **first** word of that line. After so much time of trying, he was able to remember the **first** letter. Can you help him remember **the rest of the word**?

Input

The only line of input contains the first letter of the first word of the magical line.

Output

Print the rest of the word.

Example

standard input	standard output
`B`	`TUNIS`

Problem L. Lucky Letters

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

We define some alphabetical letter C of some sequence of letters S to be Lucky if one of the following holds:

- It's the highest letter in S, i.e there is no other letter D such that $D > C$.
- It's the lowest letter in S, i.e there is no other letter D such that $D < C$.
- Both the letters directly after and before C in the alphabet exist in S. For example, if C is "c" then both "b" and "d" must exist in S.

You will be given a sequence of lowercase Latin letters, can you tell if all the letters in S are Lucky?

Input

The first line of input contains a single integer T $(1 \le T \le 1000)$ — the number of testcases. Each testcase is described using two lines.

The first line of each testcase contains a single integer N $(1 \le N \le 10^5)$ — the length of the seqeuence S.

The second line of each testcase contains N lowercase Latin letters — the seqeuence S

It is guaranteed that the sum of N over all testcases doesn't exceed 10^5.

Output

For each testcase print "YES" if the sequence S is lucky otherwise print "NO".

You can print the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

Example

standard input	standard output
3 5 abcde 3 bce 5 aaacc	YES NO YES

Problem M. K-almost palindrome

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

A **K-almost palindrome** is a string that can be made into a palindrome by shifting it to the left at most K times. For example, the string "*aabb*" is a 1-almost palindrome, a 2-almost palindrome, a 3-almost palindrome... because it needs be shifted at least 1 time to become the palindrome "*abba*". However, the string "*abcd*" is not a **K-almost palindrome** because no amount of shifting can make it a palindrome.

You are given a string of N lowercase Latin letters S. Some letters are missed and are given as ?.

Consider all possible ways of replacing each missed letter with some lowercase Latin letter, what is the number of **K-almost palindrome** resulting strings?

Input

The first line contain two integers N $(2 \leq N \leq 200)$ and K $(0 \leq K \leq N)$, N is always even.

The second line contain a string S.

Output

The number of **K-almost palindrome strings** resulting sequences mod $10^9 + 7$.

Examples

standard input	standard output

`4 1`
`aabb`
	1

`8 3`
`aa??c?b?`
	27

`6 2`
`??????`
	52676

Note

If a string is a x-almost palindrome string and a y-almost palindrome string and $x, y \leq K$, its only counted once.