A. Is it a pattern?

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Jupiter likes to find patterns. She got a birthday gift of many strings and she wants to find if the strings are *pattern-j* or not.

A string s is said to be *pattern-j* if it satisfies either of the 2 conditions:

- The string has length 0 or 1.
- The string s is a palindrome, and the string formed by the first $\left\lfloor \frac{|s|}{2} \right\rfloor$ and the last $\left\lfloor \frac{|s|}{2} \right\rfloor$ characters of s are both *pattern-j*.

Here, $|s|$ refers to the length of s and $\lfloor . \rfloor$ refers to the floor/greatest integer function.

A string is said to be a palindrome if it is equal to its reverse.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) denoting the number of test cases.

Then t line follows, each containing a string of lower-case english alphabets of atmost length 10^5.

The sum of string lengths over all test cases does not exceed 6×10^5.

Output

For each test case, output `Yes` if the string is *pattern-j* and `No` otherwise on a new line.

The grading is case-insensitive so `Yes`, `YES`, `yes` or `yEs` are all equivalent.

Scoring

Subtask #1 (20 Points): $1 \leq |s| \leq 10^3$

Subtask #2 (80 Points): $1 \leq |s| \leq 10^5$

Example

input
4 a ab abadaba abcba
output
Yes No

```
Yes
No
```

Note

1. `a` is *pattern-j* since it has length 1.
2. `ab` is not *pattern-j* since its not a palindrome.
3. The string is *pattern-j* since its a palindrome and `aba` is *pattern-j*.
4. The string is not *pattern-j* even though its a palindrome, since neither `ab` nor `ba` are *pattern-j*.

B. Game Over

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Garima found under the Christmas tree an array a of n elements and instructions for playing with it:

- At first, choose index i $(1 \le i \le n)$ - starting position in the array. Put the chip at the index i (on the value a_i)
- While $1 \le i \le n$, move the chip to the position $i + a_i$.
- If $i > n$ or $i < 1$, the game ends.

A starting position i $(1 \le i \le n)$ is considered **good** if it leads the game to an end.

For example, if $n = 5$ and $a = [4, 2, -2, -2, 3]$, then the following game options are possible:

- Garima choses $i = 1$. Game process : $i = 1 \xrightarrow{+4} 5 \xrightarrow{+3} 8$ *(end)*
- Garima choses $i = 2$. Game process : $i = 2 \xrightarrow{+2} 4 \xrightarrow{-2} 2 \xrightarrow{+2} 4$ ... *(never ends)*
- Garima choses $i = 3$. Game process : $i = 3 \xrightarrow{-2} 1 \xrightarrow{+4} 5 \xrightarrow{+3} 8$ *(end)*
- Garima choses $i = 4$. Game process : $i = 4 \xrightarrow{-2} 2 \xrightarrow{+2} 4$ ... *(never ends)*
- Garima choses $i = 5$. Game process : $i = 5 \xrightarrow{+3} 8$ *(end)*

Hence, in this case, $i = 1, 3, 5$ are good starting positions.

Help Garima by finding all the Good starting positions i $(1 \le i \le n)$.

Input

The first line contains one integer t $(1 \le t \le 10^3)$ - the number of test cases. Then t test cases follow.

The first line of each test case contains one integer n - the length of the array a.

Subtask #1 (50 points): $(1 \le n \le 10^3)$, Sum of n over all test cases does not exceed 10^3.

Subtask #2 (50 points): $(1 \le n \le 10^6)$, Sum of n over all test cases does not exceed 10^6.

The next line contains n integers $a_1, a_2, \ldots, a_n$ $(-1 \times 10^9 \leq a_i \leq 1 \times 10^9)$ - elements of the array a.

Output

For each test case, print the answer:

In the first line, print one non-negative integer k $(0 \leq k \leq n)$ - the no. of Good starting positions. If $k > 0$, print k integers in the second line - the good starting positions i $(1 \leq i \leq n)$ in ascending order.

Example

input

```
2
5
4 2 -2 -2 3
3
1 2 3
```

output

```
3
1 3 5
3
1 2 3
```

C. Poor Man's Chess

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob wanted to play chess, however, they lost all pieces except a rook. A rook moves either horizontally or vertically by any number of spaces(at least one) as long as it stays on the board.

Hence they devise a game for which they create an $N \times N$ grid. The game is played as follows:

The rook is placed initially on the top-left most square of the grid. Alice and Bob take turns alternatively to move the rook, however, they may only move the rook either right or down by any number of spaces(at least one) as long as it remains on the grid. The player who makes the rook land on the bottom-right most square of the grid loses the game. Alice makes the first move. Answer who will win the game.

Input

The first line contains the only integer N $(2 \leq N \leq 2 \cdot 10^5)$ — the dimension of the square grid.

Subtasks: **Subtask #1 (15 points):** $2 \leq N \leq 100$

Subtask #2 (85 points): Original Constraints

Output

If Alice wins the game, print "Alice", otherwise print "Bob" (without quotes).

Examples

input
2
output
Alice

input
3
output
Bob

D. Ehab Likes to Travel

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

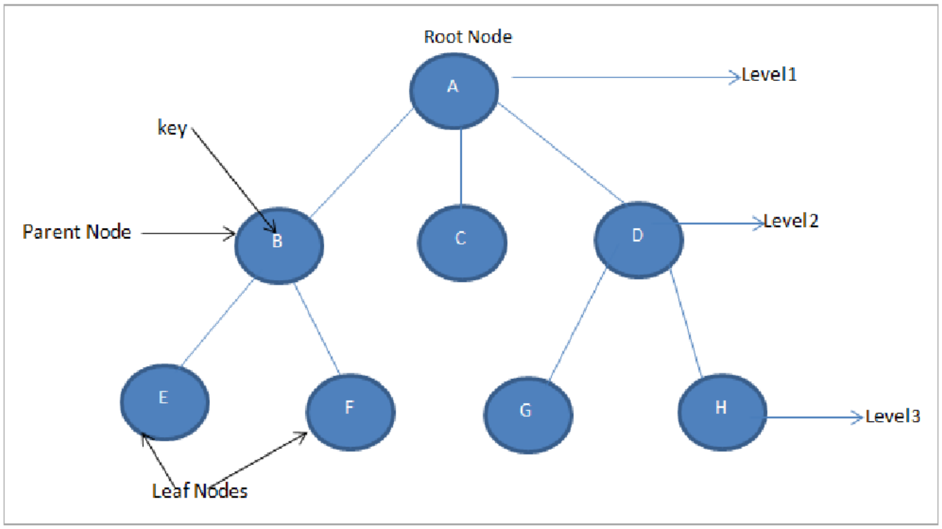
In the city of Berland, there are n checkpoints connected through $n-1$ roads. The road network is connected, i.e. can be represented as an undirected Tree where nodes represent checkpoints and edges represent roads. Length of each road is given.

Ehab wants to travel through the city. He starts at the root node (1^{st} checkpoint) and has to exit from a leaf node. Leaf is a node (other than the root node) connected through only 1 edge. However to pass through a checkpoint i, he would have to pay a tax T_i.

T_i = Sum of the length of roads on the path between 1 and i.

Suppose Ehab takes the path containing checkpoints a_1, a_2, ..., a_k, then Travelling cost = $\sum_{i=1}^{k} T_{a_i}$.

The following figure gives a clear idea about root and leaf nodes in a tree.



Ehab has a map of the city. He tries to build a strategy for optimising travelling cost. Your task is to help him find the minimum cost for travelling through Berland.

Input

The first line contains an integer t $(1 \le t \le 100)-$ the number of test cases. The description of test cases follows.

The first line of each test case contains an integer n $(2 \le n \le 10^5)-$ the number of checkpoints.

Each of the next $n-1$ lines contains three space seperated integers u, v $(1 \le u, v \le n, u \ne v)$ and l $(1 \le l \le 100)$ meaning that there is a road between checkpoints u and v and the length of this road is l.

It is guaranteed that the sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case print the $\mathbf{minimum}$ possible cost that Ehab has to pay to travel through Berland.

Scoring

$\mathbf{Subtask\ \#1\ (30\ points):}$ Sum of n over all test cases doesn't exceed 1000.

$\mathbf{Subtask\ \#2\ (70\ points):}$ Original Constraints

Example

input

```
2
3
1 2 3
1 3 2
5
1 2 1
```

```
2 3 2
2 4 3
1 5 5
```

output

```
2
4
```

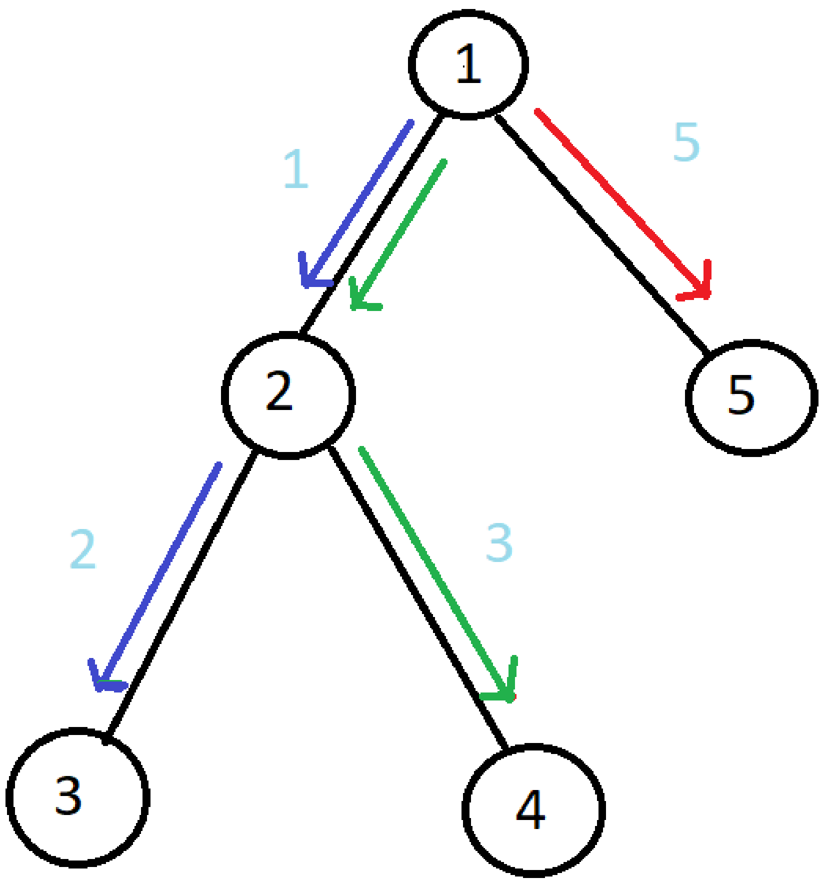
Note

EXPLANATION

Case #1: $\mathbf{T} = \{0, 3, 2\}$

Optimal path is $1 \rightarrow 3$ with Total cost = 2.

Case #2:



T_1 = 0; T_2 = 1; T_3 = 1 + 2 = 3; T_4 = 1 + 3 = 4; T_5 = 5

$\mathbf{T} = \{0, 1, 3, 4, 5\}$

Travelling Cost for Path 1(Blue) = T_1 + T_2 + T_3 = 4; for Path 2(Green) = T_1 + T_2 + T_4 = 5; for Path 3(Red) = T_1 + T_5 = 5

Optimal path is $1 \rightarrow 2 \rightarrow 3$ with Total cost = 4.

E. Valuable Strings

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Lannister is a greedy boy who is always in search for treasures. He encounters a long string (containing only lowercase English alphabets). He only considers vowels {'a', 'e', 'i' , 'o' , 'u'} as valuable characters . He wants to take back with him a valuable substring. A valuable substring is one which contains at least one valuable character.

A substring is a string which is formed from original string by removing some (possibly zero) characters from the beginning and/or end of the original string.

Despite being wealthy, he is not that smart. He wants you to figure out for him in how many ways he can take back **one** valuable string with him.

NOTE: Substrings, starting or ending at different indices of the original string are considered different even if they are lexicographically equal. (See notes for more clarification)

Input

The first line of input contains number of testcases $t (1 \le t \le 50)$. Then t lines follow. Each line containing a lower case english alphabet string whose length does not exceed 10^5

Subtask #1(10 points): length of string does not exceed 400

Subtask #2(30 points): length of string does not exceed 2×10^3

Subtask #3(60 points): length of string does not exceed 10^5

It is guaranteed that the sum of lengths of strings over all testcases is $\le 10^5$

Output

Print t lines. In each line print only a single integer, the number of ways Lannister can take back one valuable substring with him.

Examples

input
1 aa
output
3

input
1 bed

output
4

Note

Explanation Test Case 1: aa has three substrings containing vowels namely "aa", "a", "a" Notice that "a" is considered twice since they represent two different substrings (start at a different position) even though are equal lexicographically

Explanation Test Case 2 : "bed" has 4 valuable substrings "e" , "be" , "ed" , "bed".

F. Study Groups

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n seats arranged in a row, such that some of them are empty while the rest are occupied by students (A maximum of 1 student can occupy 1 seat). Now, their teacher would like to divide the class into study groups of k-students each. It's guaranteed that the total number of students in the class is a multiple of k.

To divide the class into study groups, the teacher decides to partition the seats available such that every partition has exactly k seats occupied, and each seat is a part of exactly 1 partition. A partition line can be drawn between 2 adjacent seats, and 2 partition lines exist by default: one before the 1^{st} and one after the n^{th} seat.

Your task is to find the total number of ways their teacher can partition the class. 2 ways are considered different if a pair of seats has a partition line between them in one of the ways but not in the other.

Since the answer can be very large, print it modulo $10^9 + 7$.

Input

The first line of input contains 2 space-separated integers, n and k — the total number of seats available and the number of students in each study group.

Subtask #1 (35 points) : $1 \leq k \leq n \leq 10^3$

Subtask #2 (65 points) : $1 \leq k \leq n \leq 10^5$

The next line contains n space-separated integers $a_1, a_2, \ldots, a_n$, such that $a_i = 0$ if the i^{th} seat is empty and $a_i = 1$ if the i^{th} seat is occupied by a student.

It is guaranteed that $\sum_{i=1}^{i=n} a_i$ is divisible by k.

Output

Output a single integer — The number of ways to partition the class under the given constraints modulo 1000000007.

Examples

input
3 1 1 1 1
output
1

input
8 2 1 0 1 0 0 1 1 0
output
3

Note

For the first example, you can partition the class in only 1 way, such that all partitions have exactly 1 student:

| 1 | 1 | 1 |

For the second example, you can partition the class in the following 3 ways under the given constraints:

| 1 0 1 | 0 0 1 1 0 |

| 1 0 1 0 | 0 1 1 0 |

| 1 0 1 0 0 | 1 1 0 |

G. Is time travel possible?

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Hououin Kyōma was experimenting in his Future Gadget Laboratory when he realized he had built accidentally a time machine capable of sending text messages to the past. He called these text messages D-mails. In order to send D-mails he had to input two integers p and q in his time machine. However only those d-mails were sent successfully in the past which satisfies the following condition : $p^4 - q^4 = 8k$, where k is an integer.

Given an array of n integers - $a_1, a_2, \ldots, a_n$, help Kyōma find the total number of pairs of $(a_i, a_j), i < j$ for which the D-mails were successfully sent in the past.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^5)$
.

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

Next line contains n single-spaced integers $a_1, a_2, \ldots, a_n (1 \le a_i \le 10^9)$.

The input is chosen such that the sum of n over all testcases is less than 10^5.

Subtask 1 : $(1 \le t \le 20)$, $(1 \le n \le 100)$, $(1 \le a_i \le 1000)$. Points $= 20$

Subtask 2 : $(1 \le t \le 50)$, $(1 \le n \le 200)$, $(1 \le a_i \le 10^9)$. Points $= 20$

Subtask 3 : Original Constraints. Points $= 60$

Output

For each test case, print a single integer depicting the total number of pairs for which the D-mails were sent successfully.

Example

input

```
2
3
1 2 3
3
1 3 3
```

output

```
1
3
```

Note

In the example, for the first test case $1^4 = 1$, $2^4 = 16$, $3^4 = 81$. Therefore output will be 1 as only one pair $(1, 3)$ satisfies the condition as $81 - 1 = 8 * 10$.

In the second test case, $1^4 = 1$, $3^4 = 81$. Therefore the output will be 3 as pairs $(1, 3)$, $(1, 3)$, $(3, 3)$ satisfies the condition as $1 - 81 = 8 * (-10)$, $81 - 1 = 8 * 10$ and $81 - 81 = 8 * 0$ respectively.