

Pile of books

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 64 megabytes

Davide purchased N books in preparation for the new academic year. He carefully placed them on a table in his study room.

After a few minutes, he noticed that the books were arranged in ascending order, with the smallest book at the bottom and the largest book at the top.

Realizing this, Davide decided that he preferred the books to be in a different order.

He wants to reverse their position, so the largest book is at the bottom and the smallest book is at the top.

Davide can perform the following operation a maximum of T (10000) times: he can remove K books from the top of pile A and place them, in the same order, on top of pile B ($0 \leq A, B \leq 10^9$) (he may choose to utilize a new empty pile, if desired).

The cost of the operation will be 1 if the movement of books results in pile A having less books than pile B prior to the movement, and 0 otherwise.

Help Davide to reverse the order of the books while minimizing the total cost.

Input

The first line contains the only integer N ($1 \leq N \leq 1000$), the number of books.

Your program will be tested against several test cases grouped in subtasks. The score in each subtask will be calculated as the minimum score obtained in any of its test cases, multiplied by the value of the subtask.

The score of a test case is 0 if the answer is wrong or invalid, otherwise let C be the total cost of the operations performed by your program. It can be proved that with the given constraints the minimal cost is at least 1, and a solution always exists with at most T operations. Your score will be calculated based on the following formula:

score =

$$\min\left(1 - \frac{\log_2(C) - 1}{\log_2(T) - 1}, 1\right)$$

.

Output

For each operation, output a line containing three integers K , A and B .

Examples

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

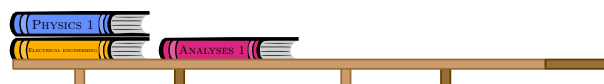
Note

In the **first sample case** initially there are 3 books on pile 0.

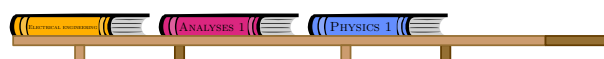


We perform the following operations:

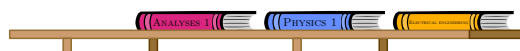
- Move 1 book from pile 0 to pile 1 with cost 0.



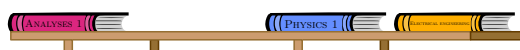
- Move 1 book from pile 0 to pile 2 with cost 0.



- Move 1 book from pile 0 to pile 3 with cost 0.



- Move 1 book from pile 1 to pile 0 with cost 0.



- Move 1 book from pile 2 to pile 0 with cost 1.



- Move 1 book from pile 3 to pile 0 with cost 1.



Finally, we have 3 books on pile 0 in the correct order, with a total cost of 2.

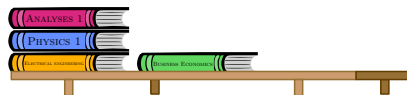
Note that for 3 books there exists a solution with cost 1, too, however according to the scoring formula, the solution with cost 2 gets a full score as well.

In the **second sample case** initially there are 4 books on pile 0.



We perform the following operations:

- Move 1 book from pile 0 to pile 1 with cost 0.



- Move 1 book from pile 0 to pile 1 with cost 0.



- Move 1 book from pile 0 to pile 2 with cost 0.



- Move 1 book from pile 0 to pile 2 with cost 1.



- Move 2 books from pile 2 to pile 1 with cost 1.



Finally, we have 4 books on pile 1 in the correct order, with a total cost of 2.

Spring Carnival

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 64 megabytes

The Spring Carnival has arrived and n tourists have reached to the event. We know that all the tourists must wear a hat which is in one of the k colors.

The carnival organizers have given each tourist a hat and asked them to sit in a line to take a group picture. However, the picture was lost, but fortunately an observer recorded for each person how many people in front of them had a hat of the same color as the hat they were wearing.

For example, if the array is $[0, 1, 0, 1, 2, 3]$, the second person had one person in front of them with the same hat color, while the fifth person had two people with the same hat color in front of them.

Assuming that the carnival organizers have an unlimited number of hats of each of the k colors, find out how many different ways exist which could have been used to assign hats to people such that the array the observer recorded would be the same as the array we would record for that given configuration?

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

Input

The input will contain on the first line n and k ($1 \leq n \leq 2 * 10^5$), ($1 \leq k \leq 10^8$).

The next line will contain the array v ($0 \leq v_i < i$).

For tests worth 10 points, ($1 \leq n, k \leq 8$).

For tests worth 30 more points, ($1 \leq n, k \leq 1000$).

Output

The output will contain a single number, representing the answer modulo $10^9 + 7$.

Examples

standard input	standard output
6 4 0 1 0 1 2 3	24
4 3 0 1 0 0	6

Note

For the second sample test case, the arrays we can generate are:

$[1, 1, 2, 3]$, $[1, 1, 3, 2]$, $[2, 2, 1, 3]$, $[2, 2, 3, 1]$, $[3, 3, 1, 2]$, $[3, 3, 2, 1]$

George Challenges the Dragon

Input file: standard input
Output file: standard output
Time limit: 0.5 seconds
Memory limit: 64 megabytes

Evil George has kidnapped the princess. Now Drogo the dragon comes to rescue her. George prefers challenging the dragon intellectually rather than fighting him, so he tells Drogo:

“I will free the princess if you can solve the following puzzle. I tell you two positive numbers N and D , and you have to tell me two different positive N -digit numbers A and B so that their greatest common divisor is D .”

Can you help the dragon to solve this challenge and save the princess? Beware, George might be so evil that there is no possible answer!

Note: the greatest common divisor of two positive integer numbers A and B is the greatest integer k such that both A and B are multiples of k .

Input

The input has two lines, both of them contain a single integer. The first line contains N ($1 \leq N \leq 18$), the second line contains D ($1 \leq D \leq 10^9$).

For test cases worth 8 points, ($1 \leq N \leq 3$).

For test cases worth 16 more points, ($1 \leq N \leq 7$).

For test cases worth 7 more points, ($D = 1$).

For test cases worth 11 more points, ($D \leq 100$).

Output

You need to write two different positive integers A and B separated by a space. Both A and B must have N digits and their greatest common divisor must be D . If there are no such numbers, you should write 0 0 to the output. If there are multiple possible solutions, you can print any of them.

Examples

standard input	standard output
3 9	180 729
6 666666	0 0

Note

In the **first sample case**, Drogo has to tell two different 3-digit numbers whose greatest common divisor is 9. There are many possible solutions, e.g. 180 and 729.

In the **second sample case**, there is only one 6-digit number which is divisible by 666666, it is 666666 itself, which means that it is not possible to give two different 6-digit numbers whose greatest common divisor is 666666.

Lottery tickets

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 64 megabytes

Tommaso would like to buy a new computer but unfortunately he doesn't have enough money. So he decided to try his luck and buy some lottery tickets.

The lottery tickets are numbered, and one of his friends told him the secret to win the lottery: in a winning ticket the sum of the digits of the ticket number is exactly N and the ticket number does not contain 3 consecutive equal digits.

When Tommaso arrived to the lottery shop, he saw that some tickets were already sold. The current ticket number is S and Tommaso must buy the tickets in order (first ticket number S , then ticket number $S + 1$, then ticket number $S + 2$, and so on).

He wants to buy some tickets until he finds a winning one. Note that it is possible that Tommaso is not able to buy a winning ticket.

What is the number of the winning ticket bought by Tommaso?

Help Tommaso to find the answer in T different scenarios.

Input

The first line contains a single integer T ($1 \leq T \leq 200$), denoting the number of test cases.

The following T lines contains 2 integers: N and S ($1 \leq N \leq 80000$), ($1 \leq |S| \leq 10000$).

For tests worth 30 points, $N \leq 32$ and the length of each S does not exceed 4.

For tests worth 30 more points, $N \leq 2000$ and the length of each S does not exceed 300.

Output

You need to write T lines with an integer: the number of the winning ticket bought by Tommaso, or -1 if he can not buy one.

Example

standard input	standard output
4	502
7 502	-1
1 123	109
10 99	4453
16 4440	

Note

In the **first sample case**:

In the first test Tommaso can buy the first ticket since it is a winning ticket.

In the second test case Tommaso can not buy any winning ticket. Note that the ticket number 1000 is not a winning ticket since it contains three consecutive 0.

Planting trees

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 64 megabytes

Since spring is coming and thus the tree planting season is about to start as well, Alice and Bob decided to start planning their future adventures.

One day, Bob came with the idea that they could spend their time together planting trees in the magic forest. Since this is a magic forest, the trees they can plant are in fact binary trees!

However, they are not just any kind of binary tree, but rooted binary trees with root at node 1 and with the property that there are more nodes on the left side of the tree than on the right side of the tree.

This spring they decided that the trees they are going to plant are of size N . Since they want to spend as much time as possible in the magic forest, let them know how many distinct binary trees with N nodes exist such that there are more nodes on the left side of the tree than on the right side of the tree.

Since the answer is big, print it modulo $10^9 + 7$.

A binary tree is a rooted tree with the property that each node has at most two children. Two binary trees are different if they have either a different set of edges or a child is the right child instead of the left child. This means that even though trees can be isomorphic, we will still count each tree as separate.

Input

The first line of the input will contain an **even** integer n , the number of nodes the trees must have ($1 \leq n \leq 10^6$).

For tests worth 10 points, ($1 \leq n \leq 10$).

For tests worth 40 more points, ($1 \leq n \leq 2000$).

Output

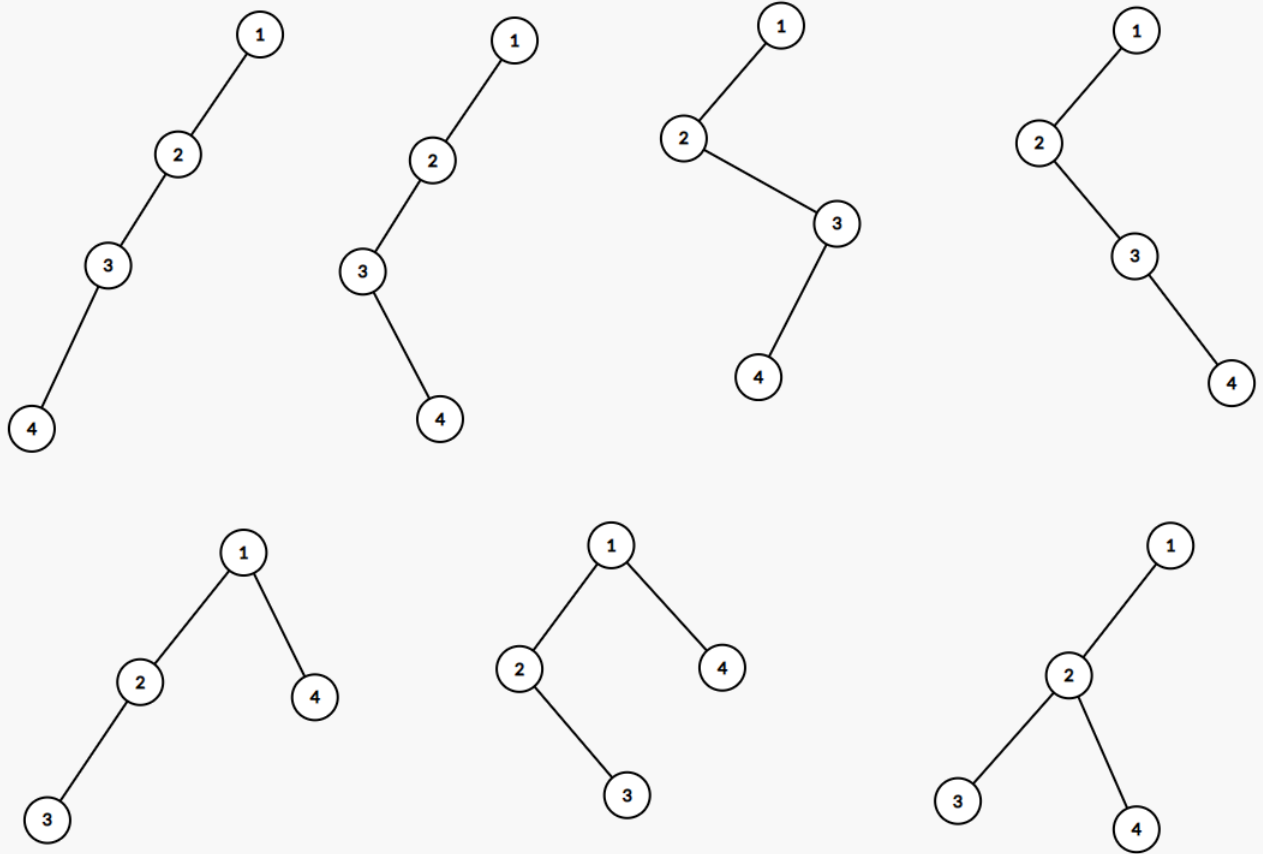
The output will contain the required answer, modulo $10^9 + 7$.

Example

standard input	standard output
4	7

Note

On the image below, the trees we can create with $N = 4$ nodes are shown in the drawing. Take note that the condition for the number of nodes from left and right subtrees must be respected only by node 1 and two trees are also different if we place a vertex on the left or on the right end of a parent subtree, in case the parent has only one adjacent vertex.



Tree Median

Input file: **standard input**
 Output file: **standard output**
 Time limit: 1.5 seconds
 Memory limit: 64 megabytes

You are given a tree with N nodes, numbered from 1 to N . The edges are numbered from 1 to $N - 1$, and edge i has a weight w_i .

Consider a simple path connecting two different nodes. If the weights on this path are $w_{i_0} \leq w_{i_1} \leq \dots \leq w_{i_k}$ (not necessarily in this order), then we define its median as $w_{i_{\lfloor k/2 \rfloor}}$.

Let M be the list of medians of all such simple paths (that is, $|M| = \frac{N(N-1)}{2}$). What is the K th smallest element of M ?

Input

The first line contains two integers, N and K ($1 \leq N \leq 50000$), ($1 \leq K \leq N * (N - 1) / 2$). The i -th of the following $N - 1$ lines contains three integers u_i , v_i and w_i ($1 \leq u_i, v_i \leq N$), ($1 \leq w_i \leq 10^9$), corresponding to an edge of weight w_i between nodes u_i and v_i .

For tests worth 8 points, $N \leq 100$.

For tests worth 19 more points, $N \leq 1000$.

For tests worth 24 points, The tree is a bamboo (no node has a degree greater than 2).

Output

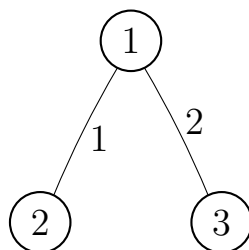
The first and only line must contain a single number: the unique integer that solves this task.

Examples

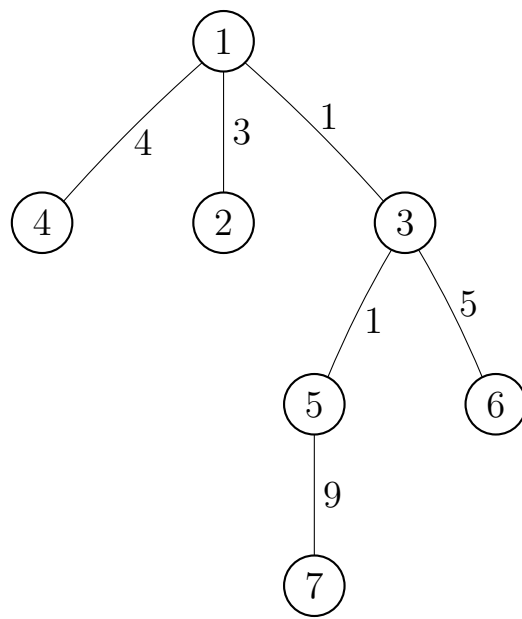
standard input	standard output
3 3 1 2 1 1 3 2	2
7 15 1 2 3 1 3 1 1 4 4 3 5 1 3 6 5 5 7 9	3

Note

In the **first sample case**, the elements of M in increasing order are 1, 1 and 2.



In the **second sample case**, the elements of M in increasing order are 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 4, 4, 5, 5 and 9.



Ancient Text

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 64 megabytes

Ancient viking people carved stories on stone using their own symbols. Björn found N versions of the same story and he already converted them to texts containing only the lowercase letters **a-z**. The texts are indexed from 0 to $N - 1$, let's denote them with $S_0, S_1, \dots, S_{N-1}$. All the texts have the same length K , but they might differ on some positions.

Now Björn wants to know which is the original version of the story, because he thinks that the others are just mutated copies of it. He suspects that the original text is the one that has the minimal average distance from all the others. We define the distance between two texts as the number of positions where they differ.

More formally, the distance between two texts S_i and S_j denoted by $dist(S_i, S_j)$ is the number of different k indices, for which $0 \leq k \leq K - 1$ and $S_i[k] \neq S_j[k]$. We are looking for the text S_{orig} for which the value $avgdist(S_i) =$

$$\frac{1}{N-1} \sum_{j=0}^{N-1} dist(S_i, S_j)$$

is minimal. (Note that $dist(S_i, S_i) = 0$). If there are multiple texts with the same average distance from the others, you should choose the one which has the smallest index of those.

Can you help Björn to determine which one is the original text?

Input

The first line contains two integers N and K ($1 \leq N, K \leq 10^5$), ($1 \leq N * K \leq 10^6$). The following N lines each contain a string S_i of length K containing only lowercase English letters "**abc...z**".

For tests worth 4 points, $N = 2$.

For tests worth 11 more points, $N = 3$.

For tests worth 21 more points, $1 \leq N, K \leq 100$.

For tests worth 25 more points, S_i consists of only letters a and b .

Output

You need to write a single line with an integer between 0 and $N - 1$, the index of the original text. If there are multiple possible solutions, print the one with the smallest index.

Examples

standard input	standard output
3 3 aab aba aaa	2
5 7 abcdefg abcdefgh abcdefgh abcdef i abcdef j	1

Note

In the **first sample case**,

- $avgdist(aab) = \frac{1}{2}(dist(aab, aba) + dist(aab, aaa)) = \frac{1}{2}(2 + 1) = 1.5$
- $avgdist(aba) = \frac{1}{2}(dist(aba, aab) + dist(aba, aaa)) = \frac{1}{2}(2 + 1) = 1.5$
- $avgdist(aaa) = \frac{1}{2}(dist(aaa, aab) + dist(aaa, aba)) = \frac{1}{2}(1 + 1) = 1$

So, the last text has the minimal average distance from the others, hence the answer is its index, which is 2.

In the **second sample case**, `abcdefh` has an average distance of 0.75 while the other texts have an average distance of 1, so the solution is the index of the first occurrence of `abcdefh`, which is 1.