A. High Load

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Arkady needs your help again! This time he decided to build his own high-speed Internet exchange point. It should consist of n nodes connected with minimum possible number of wires into one network (a wire directly connects two nodes). Exactly k of the nodes should be exit-nodes, that means that each of them should be connected to exactly one other node of the network, while all other nodes should be connected to at least two nodes in order to increase the system stability.

Arkady wants to make the system as fast as possible, so he wants to minimize the maximum distance between two exit-nodes. The distance between two nodes is the number of wires a package needs to go through between those two nodes.

Help Arkady to find such a way to build the network that the distance between the two most distant exit-nodes is as small as possible.

Input

The first line contains two integers n and k ($3 \le n \le 2 \cdot 10^5$, $2 \le k \le n - 1$) — the total number of nodes and the number of exit-nodes.

Note that it is always possible to build at least one network with n nodes and k exit-nodes within the given constraints.

Output

In the first line print the minimum possible distance between the two most distant exit-nodes. In each of the next $n - 1$ lines print two integers: the ids of the nodes connected by a wire. The description of each wire should be printed exactly once. You can print wires and wires' ends in arbitrary order. The nodes should be numbered from 1 to n. Exit-nodes can have any ids.

If there are multiple answers, print any of them.

Examples

input
```
3 2
```
output
```
2
1 2
2 3
```

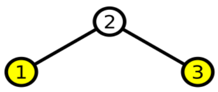
input
```
5 3
```
output
```
3
1 2
2 3
3 4
3 5
```

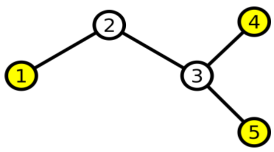
Note

In the first example the only network is shown on the left picture.

In the second example one of optimal networks is shown on the right picture.

Exit-nodes are highlighted.





B. DNA Evolution

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Everyone knows that DNA strands consist of nucleotides. There are four types of nucleotides: "A", "T", "G", "C". A DNA strand is a sequence of nucleotides. Scientists decided to track evolution of a rare species, which DNA strand was string s initially.

Evolution of the species is described as a sequence of changes in the DNA. Every change is a change of some nucleotide, for example, the following change can happen in DNA strand "AAGC": the second nucleotide can change to "T" so that the resulting DNA strand is "ATGC".

Scientists know that some segments of the DNA strand can be affected by some unknown infections. They can represent an infection as a sequence of nucleotides. Scientists are interested if there are any changes caused by some infections. Thus they sometimes want to know the value of impact of some infection to some segment of the DNA. This value is computed as follows:

- Let the infection be represented as a string e, and let scientists be interested in DNA strand segment starting from position l to position r, inclusive.
- Prefix of the string $eee...$ (i.e. the string that consists of infinitely many repeats of string e) is written under the string s from position l to position r, inclusive.
- The value of impact is the number of positions where letter of string s coincided with the letter written under it.

Being a developer, Innokenty is interested in bioinformatics also, so the scientists asked him for help. Innokenty is busy preparing VK Cup, so he decided to delegate the problem to the competitors. Help the scientists!

Input

The first line contains the string s ($1 \le |s| \le 10^5$) that describes the initial DNA strand. It consists only of capital English letters "A", "T", "G" and "C".

The next line contains single integer q ($1 \le q \le 10^5$) — the number of events.

After that, q lines follow, each describes one event. Each of the lines has one of two formats:

- `1 x c`, where x is an integer ($1 \le x \le |s|$), and c is a letter "A", "T", "G" or "C", which means that there is a change in the DNA: the nucleotide at position x is now c.
- `2 l r e`, where l, r are integers ($1 \le l \le r \le |s|$), and e is a string of letters "A", "T", "G" and "C" ($1 \le |e| \le 10$), which means that scientists are interested in the value of impact of infection e to the segment of DNA strand from position l to position r, inclusive.

Output

For each scientists' query (second type query) print a single integer in a new line — the value of impact of the infection on the DNA.

Examples

input

```
ATGCATGC
4
2 1 8 ATGC
2 2 6 TTT
1 4 T
2 2 6 TA
```

output

```
8
2
4
```

input

```
GAGTTGTTAA
6
2 3 4 TATGGTG
1 1 T
1 6 G
2 5 9 AGTAATA
1 10 G
2 2 6 TTGT
```

output

```
0
3
1
```

Note

Consider the first example. In the first query of second type all characters coincide, so the answer is 8. In the second query we compare string "TTTTT..." and the substring "TGCAT". There are two matches. In the third query, after the DNA change, we compare string "TATAT..."' with substring "TGTAT". There are 4 matches.

C. Bamboo Partition

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Vladimir wants to modernize partitions in his office. To make the office more comfortable he decided to remove a partition and plant several bamboos in a row. He thinks it would be nice if there are n bamboos in a row, and the i-th from the left is a_i meters high.

Vladimir has just planted n bamboos in a row, each of which has height 0 meters right now, but they grow 1 meter each day. In order to make the partition nice Vladimir can cut each bamboo once at any height (no greater that the height of the bamboo), and then the bamboo will stop growing.

Vladimir wants to check the bamboos each d days (i.e. d days after he planted, then after $2d$ days and so on), and cut the bamboos that reached the required height. Vladimir wants the total length of bamboo parts he will cut off to be no greater than k meters.

What is the maximum value d he can choose so that he can achieve what he wants without cutting off more than k meters of bamboo?

Input

The first line contains two integers n and k ($1 \leq n \leq 100$, $1 \leq k \leq 10^{11}$) — the number of bamboos and the maximum total length of cut parts, in meters.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — the required heights of bamboos, in meters.

Output

Print a single integer — the maximum value of d such that Vladimir can reach his goal.

Examples

input

```
3 4
1 3 5
```

output

```
3
```

input

```
3 40
10 30 50
```

output

```
32
```

Note

In the first example Vladimir can check bamboos each 3 days. Then he will cut the first and the second bamboos after 3 days, and the third bamboo after 6 days. The total length of cut parts is $2 + 0 + 1 = 3$ meters.

D. Rusty String

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Grigory loves strings. Recently he found a metal strip on a loft. The strip had length n and consisted of letters "V" and "K". Unfortunately, rust has eaten some of the letters so that it's now impossible to understand which letter was written.

Grigory couldn't understand for a long time what these letters remind him of, so he became interested in the following question: if we put a letter "V" or "K" on each unreadable position, which values can the period of the resulting string be equal to?

A period of a string is such an integer d from 1 to the length of the string that if we put the string shifted by d positions to the right on itself, then all overlapping letters coincide. For example, 3 and 5 are periods of "VKKVK".

Input

There are several (at least one) test cases in the input. The first line contains single integer — the number of test cases.

There is an empty line before each test case. Each test case is described in two lines: the first line contains single integer n $(1 \le n \le 5 \cdot 10^5)$ — the length of the string, the second line contains the string of length n, consisting of letters "V", "K" and characters "?". The latter means the letter on its position is unreadable.

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

For hacks you can only use tests with one test case.

Output

For each test case print two lines. In the first line print the number of possible periods after we replace each unreadable letter with "V" or "K". In the next line print all these values in increasing order.

Example

input

```
3

5
V??VK

6
??????

4
?VK?
```

output

```
2
3 5
6
1 2 3 4 5 6
3
2 3 4
```

Note

In the first test case from example we can obtain, for example, "VKKVK", which has periods 3 and 5.

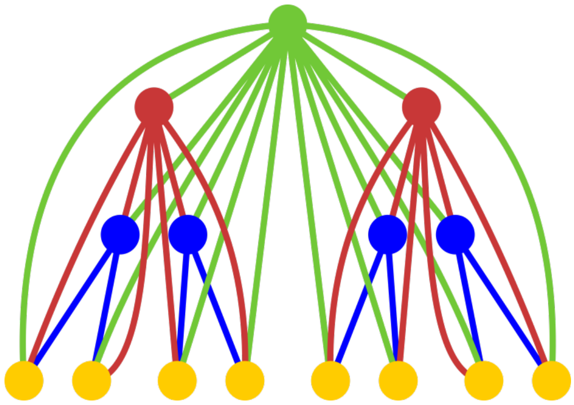
In the second test case we can obtain "VVVVVV" which has all periods from 1 to 6.

In the third test case string "KVKV" has periods 2 and 4, and string "KVKK" has periods 3 and 4.

E. Singer House

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

It is known that passages in Singer house are complex and intertwined. Let's define a Singer k-house as a graph built by the following process: take complete binary tree of height k and add edges from each vertex to all its successors, if they are not yet present.



Singer 4-house

Count the number of non-empty paths in Singer k-house which do not pass the same vertex twice. Two paths are distinct if the sets or the orders of visited vertices are different. Since the answer can be large, output it modulo $10^9 + 7$.

Input

The only line contains single integer k ($1 \le k \le 400$).

Output

Print single integer — the answer for the task modulo $10^9 + 7$.

Examples

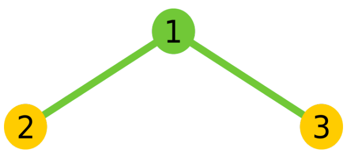
input
```
2
```
output
```
9
```

input
```
3
```
output
```
245
```

input
```
20
```
output
```
550384565
```

Note

There are 9 paths in the first example (the vertices are numbered on the picture below): `1`, `2`, `3`, `1-2`, `2-1`, `1-3`, `3-1`, `2-1-3`, `3-1-2`.



Singer 2-house

F. Perpetual Motion Machine

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Developer Petr thinks that he invented a perpetual motion machine. Namely, he has a lot of *elements*, which work in the following way.

Each element has one controller that can be set to any non-negative real value. If a controller is set on some value x, then the controller consumes x^2 energy units per second. At the same time, any two elements connected by a wire produce $y \cdot z$ energy units per second, where y and z are the values set on their controllers.

Petr has only a limited number of wires, so he has already built some scheme of elements and wires, and is now interested if it's possible to set the controllers in such a way that the system produces **at least as much** power as it consumes, and at least one controller is set on the value different from 0. Help him check this, and if it's possible, find the required **integer** values that should be set.

It is guaranteed that if there exist controllers' settings satisfying the above conditions, then there exist required integer values not greater than 10^6.

Input

There are several (at least one) test cases in the input. The first line contains single integer — the number of test cases.

There is an empty line before each test case. The first line of test case contains two integers n and m ($1 \le n \le 10^5$, $0 \le m \le 10^5$) — the number of elements in the scheme and the number of wires.

After that, m lines follow, each of them contains two integers a and b ($1 \le a, b \le n$) — two elements connected by a wire. No element is connected with itself, no two elements are connected by more than one wire.

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

For hacks you can only use tests with one test case.

Output

Print answer for each test case.

For each test case print "YES" if it's possible to set the controllers in such a way that the consumed power is not greater than the power produced, and the required values on the next line. The settings should be integers from 0 to 10^6, inclusive, and at least one value should be different from 0. If there are multiple answers, print any of them.

If it's not possible to set the controllers in the required way, print one line "NO".

Example

input

```
4

4 4
1 2
2 3
3 4
4 2

3 2
2 3
3 1

4 6
1 2
3 4
4 2
1 4
1 3
3 2

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

output

```
YES
1 2 2 1
NO
YES
1 1 1 1
YES
1 5 1 1 1 1 1 1 1 1
```

Note

In the first example it's possible to set the controllers in the required way, for example, in the following way: set 1 on the first element, set 2 on the second and on the third, set 1 on the fourth. The consumed power is then equal to $1^2 + 2^2 + 2^2 + 1^2 = 10$ energy units per second, the produced power is equal to $1 \cdot 2 + 2 \cdot 2 + 2 \cdot 1 + 2 \cdot 1 = 10$ energy units per second. Thus the answer is "`YES`".

In the second test case it's not possible to set the controllers in the required way. For example, if we set all controllers to 0.5, then the consumed powers equals 0.75 energy units per second, while produced power equals 0.5 energy units per second.

G. Dirty Arkady's Kitchen

time limit per test: 6 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Arkady likes to walk around his kitchen. His labyrinthine kitchen consists of several important places connected with passages. Unfortunately it happens that these passages are flooded with milk so that it's impossible to pass through them. Namely, it's possible to pass through each passage in any direction only during some time interval.

The lengths of all passages are equal and Arkady makes through them in one second. For security reasons, Arkady can never stop, also, he can't change direction while going through a passage. In other words, if he starts walking in some passage, he should reach its end and immediately leave the end.

Today Arkady needs to quickly reach important place n from place 1. He plans to exit the place 1 at time moment 0 and reach the place n as early as he can. Please find the minimum time he should spend on his way.

Input

The first line contains two integers n and m ($1 \le n \le 5 \cdot 10^5$, $0 \le m \le 5 \cdot 10^5$) — the number of important places and the number of passages, respectively.

After that, m lines follow, each of them describe one passage. Each line contains four integers a, b, l and r ($1 \le a, b \le n$, $a \ne b$, $0 \le l < r \le 10^9$) — the places the passage connects and the time segment during which it's possible to use this passage.

Output

Print one integer — minimum time Arkady should spend to reach the destination. If he can't reach the place n, print `-1`.

Examples

input

```
5 6
1 2 0 1
2 5 2 3
2 5 0 1
1 3 0 1
3 4 1 2
4 5 2 3
```

output

```
3
```

input

```
2 1
1 2 1 100
```

output

```
-1
```

Note

In the first example Arkady should go through important places $1 \to 3 \to 4 \to 5$.

In the second example Arkady can't start his walk because at time moment 0 it's impossible to use the only passage.