Whisperings

Práxedes wants to give Yael a postcard for Valentine's Day, but he is too embarrassed to give it to her directly. So he has decided that he will give it to a friend, the friend will give it to another friend of his, and so on until it reaches Yael.

Unfortunately for Práxedes, people at his school are very nosy and they will not miss the opportunity to discuss the situation with friends. Discreetly, of course. So, when a person (except Yael and Práxedes) receives the postcard, the *whispering level* will increase by one unit in that person and in all his friends. (Initially, the whispering level of all people is 0).

Práxedes does not want anyone to talk too much about it, so, knowing the friendship relationships of all the people in the institute, he wants to design the "route" that the postcard will follow to Yael that minimizes **the maximum** among all the whispering levels of all the people. Help Práxedes find this minimum level and a route to achieve it.

Input and output

The first line of the input contains the number of cases T.

For each case there will be an input line with two integers n, m: the number of people in the school and the number of pairs of friends. People are numbered from 0 to $n-1$ and the 0 person is Práxedes and the $n-1$ is Yael.

Then, m lines follow, each with a pair of integers u, v, denoting a (symmetric) friendship relationship between persons u and v.

For each case, if there is a way to get the postcard to Yael, you must print a line with two integers c, l, where c is the minimum maximum whispering level that is possible to reach and l is the number of intermediary people (not including Práxedes or Yael) that the postcard passes through on the route you have found. Then you must print a second line with l integers: the l intermediary people on your route. If there were several routes that minimize the maximum whispering level, you can print any one of them. In case there is no way to get the postcard to Yael, you must print a single line with two integers `0 0`.

0.1 Example

Input:

```
3
2 0
3 2
0 1
1 2
6 7
0 1
1 2
1 3
3 0
3 4
4 2
4 5
```

Output:

```
0 0
1 1
1
2 2
3 4
```

Constraints

$1 \leq T \leq 5000$.

$2 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$.

The sum of n for all cases and the sum of m for all cases are at most $2 \cdot 10^5$.

$0 \leq u, v \leq n-1$, $u \neq v$, no unordered pairs (u, v) will be repeated in the input.

It is guaranteed that there will be no friendship relationship between 0 and $n-1$.

Subtasks

1. (12 points) $n \leq 3$.
2. (20 points) $n \leq 8$.
3. (19 points) $m = n-1$, for the i-th (u, v) pair given in the entry $(1 \leq i \leq m)$ it is satisfied that $u = i$ and $v < i$.
4. (49 points) No additional constraints.

Goulash

Szeged goulash is a braised meat dish originating from Szeged, Hungary, the site of IOI 2023. But Félix likes the goulash they make in the Czech Republic better, because it has *knedlíky*.

Félix wants to eat exactly N knedlíky. There are A restaurants that offer goulash with 4 knedlíky, B with 5 knedlíky, and C with 6 knedlíky. If Félix can go eat at some of those restaurants and does not want to repeat at any restaurant, is it possible for him to get to eat exactly N knedlíky?

Formally, you must say whether there exist integers $0 \leq x \leq A, 0 \leq y \leq B, 0 \leq z \leq C$ such that $4x + 5y + 6z = N$.

Input and output

The first line of the input contains the number of cases T.

For each case there will be one input line with four integers N, A, B, C.

For each case, you must print a line with `SI` if Félix can eat N knedlíky and `NO` otherwise.

Example

Input:

```
4
0 1 2 3
26 1 2 3
4 0 0 0
11 2 1 0
```

Output:

```
SI
SI
NO
NO
```

Explanation: In the first case, Félix can eat 0 knedlíky by going to no restaurants at all. In the second, he can go to a restaurant of the first type, 2 of the second type, and 2 of the third type to eat $1 \cdot 4 + 2 \cdot 5 + 2 \cdot 6 = 26$ knedlíky.

Constraints

$1 \leq T \leq 100\,000$.

$0 \leq N, A, B, C \leq 10^8$.

Subtasks

.

1. (20 points) $A, B, C \leq 100$.
2. (19 points) $N, A, B, C > 100$.
3. (18 points) $B = 0$.
4. (17 points) $A \cdot B \cdot C = 0$.
5. (26 points) No additional constraints.

Valuable List

Given a list of n positive integers $L = (a_1, \ldots, a_n)$, Ana defines its value $V(L)$ as:

$$V(L) = n + \sum_{i=1}^{n-1} |a_i - a_{i+1}| = n + |a_1 - a_2| + |a_2 - a_3| + \ldots + |a_{n-1} - a_n|$$

Given v and k, help Anne find the k-th list in lexicographic order that has value equal to v.

Recall that, given two lists $L_1 = (a_1, \ldots, a_n)$ and $L_2 = (b_1, \ldots, b_m)$, L_1 comes before L_2 in lexicographic order if:

- There exists a $j \leq \min(n, m)$ such that $a_i = b_i$ for all $1 \leq i < j$ and $a_j < b_j$, or:
- $a_i = b_i$ for all $1 \leq i \leq n$ and $n < m$.

Input and output

The first line of the input contains the number of cases T.

For each case there will be an input line with two integers v, k.

For each case you must print a line with the integers of the requested list separated by spaces.

Example

Input:

```
5
5 3
1 1
2 4
3 10
4 1000000000000000
```

Output:

```
1 1 2 2
1
4 4
4 4 4
142857142857144 142857142857144 142857142857143
```

Explanation of the first case: The 5-valued lists are, in lexicographic order:
$(1,1,1,1,1), (1,1,1,2), (1,1,2,2), (1,1,3), (1,2,1), (1,2,2,2), \ldots$.

Constraints

$1 \leq T \leq 200\,000$.

$1 \leq v \leq 10^6$.

$1 \leq k \leq 10^{15}$.

The sum of v for all cases will be at most 10^6.

Subtasks

1. (3 points) $v = 1$.
2. (10 points) $v \leq 5$.
3. (4 points) $k = 1$.
4. (8 points) $k \leq 5$.
5. (18 points) $v, k \leq 100$, the sum of $v \cdot k$ for all cases will be at most 10^6.
6. (19 points) $v \leq 1000$.
7. (20 points) The sum of k for all cases will be at most 10^6.
8. (18 points) No additional constraints.

Cyclic numbers

Empúries was a Greek city, later Roman, located near the sea in the Gulf of Roses on the Costa Brava. Nowadays they are ruins that can be visited. Pepa and Maria are two archaeologists who work there.

During the last year they have found pieces of stone with engraved numbers in decimal representation; it seems that some merchant brought them from some unknown place. From what they have observed, these numbers have a very special property. If we move the last digit to the beginning we are left with a number that is a multiple of the original.

For example, the number 142857 has this property because if we move the last digit to the beginning we are left with 714285, which is 142857×5.

The problem they have now is that they have found some stones that are incomplete, which means that there may be missing digits at the beginning and at the end of the number.

Knowing the characteristic property of the numbers represented in these stones, Pepa and Maria want to know what is the minimum number of digits they have to add before or after the numbers found to get a number with the desired property. If it is not possible to achieve this, they also want to know, since this would mean that their investigation is on the wrong track.

Following the previous example, if we find a stone with the number 4285, the answer would be 2 since if we add a 1 in front and a 7 behind we get the number 142857.

(We consider that a valid number has no leading zeros).

Input and output

Each file starts with an integer T, the number of cases.

Then follow T lines each with an integer k representing the number inscribed in a stone.

For each case, write a line with the number of digits that must be added as a minimum to get a number with the desired property or "No" if it is not possible to get it.

Examples

Example 1:

Input:

```
4
142856
142857
230768
230769
```

Output:

```
No
0
No
0
```

Example 2:

Input:

```
4
5517
9831508
4745762
7128
```

Output:

```
24
No
51
No
```

Constraints

$1 \leq T \leq 10^5$.

$1 \leq k \leq 10^{1000000}$.

Let S be the sum of the number of digits of k for all cases. Then $S \leq 1\,000\,000$.

Subtasks

1. (5 points) The answer will be 0 or "No" and $k \leq 10^{15}$.
2. (10 points) $k \leq 10^{15}$.
3. (4 points) The answer will be 0 or "No" and $S \leq 1000$.
4. (32 points) $S \leq 1000$.
5. (14 points) The answer will be 0 or "No".
6. (35 points) No additional restrictions.

Prison

Prisoners in Szeged prison want to take advantage of their city being distracted by the IOI to escape from jail. They find themselves in the room where the prison's keys are kept, but they don't know which key opens the only exit door.

There are n keys in a row, leaving 1 meter between keys, and they have a device that, when asked about one of the keys, answers them with a d number, the distance to the key that opens the prison. But the device has been infected by the spirit of the IOI, now it will answer with $d \,\&\, x$, where $0 \le x \le \frac{n}{2}$ is an integer they don't know.

The operation & (*AND* bit by bit) of two non-negative integers is calculated as follows: the two numbers are written in binary and the result is the number that written in binary has 1 in the positions in which both operands have a 1, and 0 in the positions in which one of the operands has a 0.

For example, $26\&14 = 10$, because 26 in binary is 11010, 14 in binary is 1110, and the number resulting from putting 1 in the positions in which both operands have a 1 is 1010, which if we write it in decimal base is 10.

$$\begin{array}{r} 1\,1\,0\,1\,0 \\ \&\,0\,1\,1\,1\,0 \\ \hline 0\,1\,0\,1\,0 \end{array}$$

Manuel, the ringleader of the operation, has also been infected by the spirit of the IOI and has set himself a new problem: to find the number x. So he asks you to create a program that, given n, analyzes a series of keys with the device and responds with the value of x (the location of the key is irrelevant). Since the device takes a long time to perform the calculations, Manuel wants to reduce the number of questions he asks for fear of being discovered by the guards.

Input and output

This is an interactive problem. You must refresh the output each time you print data (`cout « endl` or `cout « flush` in C++, `System.out.flush()` in Java, `stdout.flush()` in Python).

The first line of the input contains the number of cases T.

For each case, you must first read a line with the number n. After reading this value, you must ask the interactor questions:

To ask a question, you must print a line with the format `? i`, where $0 \le i < n$ is the key index (note that it starts at 0). You will then be answered with a line with the number $d \,\&\, x$ written in decimal. If you ask an invalid question or exceed the question limit, you will be answered with `-1`, and in that case your program must terminate immediately.

To give the answer, you must print a line in the format `! x`, where $0 \le x \le \frac{n}{2}$ is the value of x that you have determined. Next, you must read an answer with one character, which will be:

- `+`, if you go to the next case. In that case your program should read the new integer n next and continue with the new case.
- `-`, if you do not go to the next case, either because the cases are finished or because you have given an incorrect answer. In that case, your program must terminate immediately.

Example

Here we show an example interaction in which, for clarification, the value of x and the k position of the key are shown for each case.

Judge	Program	Commentary
2		
3		$x = 1, k = 0$
	? 0	
0		$\lvert 0 - 0 \rvert \,\&\, 1 = 0$
	? 1	
1		$\lvert 1 - 0 \rvert \,\&\, 1 = 1$
	! 1	
$+$		
5		$x = 2, k = 1$
	? 0	
0		$\lvert 0 - 1 \rvert \,\&\, 2 = 0$
	? 1	
0		$\lvert 1 - 1 \rvert \,\&\, 2 = 0$
	? 2	
0		$\lvert 2 - 1 \rvert \,\&\, 2 = 0$
	? 3	
2		$\lvert 3 - 1 \rvert \,\&\, 2 = 2$
	? 4	
2		$\lvert 4 - 1 \rvert \,\&\, 2 = 2$
	! 2	
$-$		

Constraints

$1 \leq T \leq 100$.

$2 \leq n \leq 10^9$.

$0 \leq x \leq \frac{n}{2}$.

You can ask at most 50 questions. Giving the answer does not count as a question.

The value of x and the key index are fixed at the beginning of the interaction, i.e., they do not change adaptively according to the questions you ask.

Subtasks

1. (10 points) $n = 2$.
2. (15 points) $n = 50$.
3. (25 points) $n = 1000$.
4. (15 points) The key that opens the prison is in position 0.
5. (20 points) The key that opens the prison is in position 0 or in position $n - 1$.
6. (15 points) No additional constraints.

Additionally, the score you get in a subtask depends on the number of questions you do: to get a full score you must do at most 4 questions and to get a positive score you must do at most 50 questions. The score of each subtask is multiplied by a multiplier $M(q)$, where q is the maximum number of questions you have asked in the cases of that subtask. The value of $M(q)$ is given by:

$$M(q) = \begin{cases} 0 & q > 50 \\ 0.8 + \frac{50-q}{250} & 50 \geq q > 4 \\ 1.0 & 4 \geq q \end{cases}$$