A. Pizza, Pizza, Pizza!!!

time limit per test: 1 second
memory limit per test: 128 megabytes
input: standard input
output: standard output

Katie, Kuro and Shiro are best friends. They have known each other since kindergarten. That's why they often share everything with each other and work together on some very hard problems.

Today is Shiro's birthday. She really loves pizza so she wants to invite her friends to the pizza restaurant near her house to celebrate her birthday, including her best friends Katie and Kuro.

She has ordered a very big round pizza, in order to serve her many friends. Exactly $$$n$$$ of Shiro's friends are here. That's why she has to divide the pizza into $$$n + 1$$$ slices (Shiro also needs to eat). She wants the slices to be exactly the same size and shape. If not, some of her friends will get mad and go home early, and the party will be over.

Shiro is now hungry. She wants to cut the pizza with *minimum* of straight cuts. A cut is a straight segment, it might have ends inside or outside the pizza. But she is too lazy to pick up the calculator.

As usual, she will ask Katie and Kuro for help. But they haven't come yet. Could you help Shiro with this problem?

Input

A single line contains one non-negative integer $$$n$$$ ($$$0 \le n \leq 10^{18}$$$) — the number of Shiro's friends. The circular pizza has to be sliced into $$$n + 1$$$ pieces.

Output

A single integer — the number of straight cuts Shiro needs.

Examples

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

input
```
4
```
output
```
5
```

Note

To cut the round pizza into quarters one has to make two cuts through the center with angle $$$90^{\circ}$$$ between them.

To cut the round pizza into five equal parts one has to make five cuts.

B. Treasure Hunt

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

After the big birthday party, Katie still wanted Shiro to have some more fun. Later, she came up with a game called *treasure hunt*. Of course, she invited her best friends Kuro and Shiro to play with her.

The three friends are very smart so they passed all the challenges very quickly and finally reached the destination. But the treasure can only belong to one cat so they started to think of something which can determine who is worthy of the treasure. Instantly, Kuro came up with some ribbons.

A random colorful ribbon is given to each of the cats. Each color of the ribbon can be represented as an uppercase or lowercase Latin letter. Let's call a consecutive subsequence of colors that appears in the ribbon a *subribbon*. The *beauty* of a ribbon is defined as the maximum number of times one of its subribbon appears in the ribbon. The more the subribbon appears, the more beautiful is the ribbon. For example, the ribbon `aaaaaaa` has the beauty of $$$7$$$ because its subribbon `a` appears $$$7$$$ times, and the ribbon `abcdabc` has the beauty of $$$2$$$ because its subribbon `abc` appears twice.

The rules are simple. The game will have $$$n$$$ turns. Every turn, each of the cats must change strictly **one** color (at one position) in his/her ribbon to an arbitrary color which is **different** from the unchanged one. For example, a ribbon `aaab` can be changed into `acab` in one turn. The one having the most beautiful ribbon after n turns wins the treasure.

Could you find out who is going to be the winner if they all play optimally?

Input

The first line contains an integer n ($0 \leq n \leq 10^{9}$) — the number of turns.

Next 3 lines contain 3 ribbons of Kuro, Shiro and Katie one per line, respectively. Each ribbon is a string which contains no more than 10^{5} uppercase and lowercase Latin letters and is not empty. It is guaranteed that the length of all ribbons are equal for the purpose of fairness. Note that uppercase and lowercase letters are considered different colors.

Output

Print the name of the winner ("`Kuro`", "`Shiro`" or "`Katie`"). If there are at least two cats that share the maximum beauty, print "`Draw`".

Examples

input

```
3
Kuroo
Shiro
Katie
```

output

```
Kuro
```

input

```
7
treasurehunt
threefriends
hiCodeforces
```

output

```
Shiro
```

input

```
1
abcabc
cbabac
ababca
```

output

```
Katie
```

input

```
15
foPaErcvJ
mZaxowpbt
mkuOlaHRE
```

output

```
Draw
```

Note

In the first example, after 3 turns, Kuro can change his ribbon into `ooooo`, which has the beauty of 5, while reaching such beauty for Shiro and Katie is impossible (both Shiro and Katie can reach the beauty of at most 4, for example by changing Shiro's ribbon into `SSiSS` and changing Katie's ribbon into `Kaaaa`). Therefore, the winner is Kuro.

In the fourth example, since the length of each of the string is 9 and the number of turn is 15, everyone can change their ribbons in some way to reach the maximal beauty of 9 by changing their strings into `zzzzzzzzz` after 9 turns, and repeatedly change their strings into `azzzzzzzz` and then into `zzzzzzzzz` thrice. Therefore, the game ends in a draw.

C. Kuro and Walking Route

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Kuro is living in a country called Uberland, consisting of n towns, numbered from 1 to n, and $n - 1$ bidirectional roads connecting these towns. It is possible to reach each town from any other. Each road connects two towns a and b. Kuro loves walking and he is planning to take a walking marathon, in which he will choose a pair of towns (u, v) ($u \neq v$) and walk from u using the shortest path to v (note that (u, v) is considered to be different from (v, u)).

Oddly, there are 2 special towns in Uberland named Flowrisa (denoted with the index x) and Beetopia (denoted with the index y). Flowrisa is a town where there are many strong-scent flowers, and Beetopia is another town where many bees live. In particular, Kuro will avoid any pair of towns (u, v) if on the path from u to v, he reaches Beetopia after he reached Flowrisa, since the bees will be attracted with the flower smell on Kuro’s body and sting him.

Kuro wants to know how many pair of city (u, v) he can take as his route. Since he’s not really bright, he asked you to help him with this problem.

Input

The first line contains three integers n, x and y ($1 \leq n \leq 3 \cdot 10^5$, $1 \leq x, y \leq n$, $x \ne y$) - the number of towns, index of the town Flowrisa and index of the town Beetopia, respectively.

$n - 1$ lines follow, each line contains two integers a and b ($1 \leq a, b \leq n$, $a \ne b$), describes a road connecting two towns a and b.

It is guaranteed that from each town, we can reach every other town in the city using the given roads. That is, the given map of towns and roads is a tree.

Output

A single integer resembles the number of pair of towns (u, v) that Kuro can use as his walking route.

Examples

input

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

output

```
5
```

input

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

output

```
4
```

Note

On the first example, Kuro can choose these pairs:

- $(1, 2)$: his route would be $1 \rightarrow 2$,
- $(2, 3)$: his route would be $2 \rightarrow 3$,
- $(3, 2)$: his route would be $3 \rightarrow 2$,
- $(2, 1)$: his route would be $2 \rightarrow 1$,
- $(3, 1)$: his route would be $3 \rightarrow 2 \rightarrow 1$.

Kuro can't choose pair $(1, 3)$ since his walking route would be $1 \rightarrow 2 \rightarrow 3$, in which Kuro visits town 1 (Flowrisa) and then visits town 3 (Beetopia), which is not allowed (note that pair $(3, 1)$ is still allowed because although Kuro visited Flowrisa and Beetopia, he did not visit them in that order).

On the second example, Kuro can choose the following pairs:

- $(1, 2)$: his route would be $1 \rightarrow 2$,
- $(2, 1)$: his route would be $2 \rightarrow 1$,
- $(3, 2)$: his route would be $3 \rightarrow 1 \rightarrow 2$,
- $(3, 1)$: his route would be $3 \rightarrow 1$.

D. Kuro and GCD and XOR and SUM

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

Kuro is currently playing an educational game about numbers. The game focuses on the greatest common divisor (GCD), the XOR value, and the sum of two numbers. Kuro loves the game so much that he solves levels by levels day by day.

Sadly, he's going on a vacation for a day, and he isn't able to continue his solving streak on his own. As Katie is a reliable person, Kuro kindly asked her to come to his house on this day to play the game for him.

Initally, there is an empty array a. The game consists of q tasks of two types. The first type asks Katie to add a number u_i to a. The second type asks Katie to find a number v existing in a such that $k_i \mid GCD(x_i, v)$, $x_i + v \leq s_i$, and $x_i \oplus v$ is maximized, where $\oplus$ denotes the bitwise XOR operation, $GCD(c, d)$ denotes the greatest common divisor of integers c and d, and $y \mid x$ means x is divisible by y, or report `-1` if no such numbers are found.

Since you are a programmer, Katie needs you to automatically and accurately perform the tasks in the game to satisfy her dear friend Kuro. Let's help her!

Input

The first line contains one integer q ($2 \leq q \leq 10^{5}$) — the number of tasks the game wants you to perform.

q lines follow, each line begins with an integer t_i — the type of the task:

- If $t_i = 1$, an integer u_i follow ($1 \leq u_i \leq 10^{5}$) — you have to add u_i to the array a.
- If $t_i = 2$, three integers x_i, k_i, and s_i follow ($1 \leq x_i, k_i, s_i \leq 10^{5}$) — you must find a number v existing in the array a such that $k_i \mid GCD(x_i, v)$, $x_i + v \leq s_i$, and $x_i \oplus v$ is maximized, where $\oplus$ denotes the XOR operation, or report `-1` if no such numbers are found.

It is guaranteed that the type of the first task is type 1, and there exists at least one task of type 2.

Output

For each task of type 2, output on one line the desired number v, or `-1` if no such numbers are found.

Examples

input

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

output

```
2
1
-1
```

input

```
10
1 9
2 9 9 22
2 3 3 18
1 25
2 9 9 20
2 25 25 14
1 20
2 26 26 3
1 14
2 20 20 9
```

output

```
9
9
9
-1
-1
-1
```

Note

In the first example, there are 5 tasks:

- The first task requires you to add 1 into a. a is now $\left\{1\right\}$.
- The second task requires you to add 2 into a. a is now $\left\{1, 2\right\}$.
- The third task asks you a question with $x = 1$, $k = 1$ and $s = 3$. Taking both 1 and 2 as v satisfies $1 \mid GCD(1, v)$ and $1 + v \leq 3$. Because $2 \oplus 1 = 3 > 1 \oplus 1 = 0$, 2 is the answer to this task.
- The fourth task asks you a question with $x = 1$, $k = 1$ and $s = 2$. Only $v = 1$ satisfies $1 \mid GCD(1, v)$ and $1 + v \leq 2$, so 1 is the answer to this task.
- The fifth task asks you a question with $x = 1$, $k = 1$ and $s = 1$. There are no elements in a that satisfy the conditions, so we report `-1` as the answer to this task.

E. Kuro and Topological Parity

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Kuro has recently won the "Most intelligent cat ever" contest. The three friends then decided to go to Katie's home to celebrate Kuro's winning. After a big meal, they took a small break then started playing games.

Kuro challenged Katie to create a game with only a white paper, a pencil, a pair of scissors and a lot of arrows (you can assume that the number of arrows is infinite). Immediately, Katie came up with the game called *Topological Parity*.

The paper is divided into n pieces enumerated from 1 to n. Shiro has painted some pieces with some color. Specifically, the i-th piece has color c_{i} where $c_{i} = 0$ defines black color, $c_{i} = 1$ defines white color and $c_{i} = -1$ means that the piece hasn't been colored yet.

The rules of the game is simple. Players must put some arrows between some pairs of different pieces in such a way that for each arrow, the number in the piece it starts from is less than the number of the piece it ends at. Also, two different pieces can only be connected by **at most** one arrow. After that the players must choose the color ($$$0$$$ or $$$1$$$) for each of the unpainted pieces. The score of a valid way of putting the arrows and coloring pieces is defined as the number of paths of pieces of alternating colors. For example, $$$[1 \to 0 \to 1 \to 0]$$$, $$$[0 \to 1 \to 0 \to 1]$$$, $$$[1]$$$, $$$[0]$$$ are valid paths and will be counted. You can only travel from piece $$$x$$$ to piece $$$y$$$ if and only if there is an arrow from $$$x$$$ to $$$y$$$.

But Kuro is not fun yet. He loves parity. Let's call his favorite parity $$$p$$$ where $$$p = 0$$$ stands for "even" and $$$p = 1$$$ stands for "odd". He wants to put the arrows and choose colors in such a way that the score has the parity of $$$p$$$.

It seems like there will be so many ways which satisfy Kuro. He wants to count the number of them but this could be a very large number. Let's help him with his problem, but print it modulo $$$10^{9} + 7$$$.

Input

The first line contains two integers $$$n$$$ and $$$p$$$ ($$$1 \leq n \leq 50$$$, $$$0 \leq p \leq 1$$$) — the number of pieces and Kuro's wanted parity.

The second line contains $$$n$$$ integers $$$c_{1}, c_{2}, ..., c_{n}$$$ ($$$-1 \leq c_{i} \leq 1$$$) — the colors of the pieces.

Output

Print a single integer — the number of ways to put the arrows and choose colors so the number of valid paths of alternating colors has the parity of $$$p$$$.

Examples

input

```
3 1
-1 0 1
```

output

```
6
```

input

```
2 1
1 0
```

output

```
1
```

input

```
1 1
-1
```

output

```
2
```

Note

In the first example, there are $$$6$$$ ways to color the pieces and add the arrows, as are shown in the figure below. The scores are $$$3, 3, 5$$$ for the first row and $$$5, 3, 3$$$ for the second row, both from left to right.