A. Crystals in making

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

Jesse Pinkman and Barney Stinson teamed up to make some bags full of crystals(inspired by rheaOP). Jesse wanted to make some extra money and Barney as usual just wanted to make his life more legendary.

Since the Expert Meth Maker Heisenberg is not around, Jesse needs to figure out on his own how to make crystals using some notes of Heisenberg. Notes are coded and Jesse needs to decode them.

The notes contain a single number k. Jesse has an infinite sequence of elements 2^0 ,2^1 ,2^2 ,2^3 ,2^4 ....and so on.To decode the notes Jesse needs to find the minimum number of elements he need to use from the sequence such that sum of them is k.

As Jesse is not good at solving problems he asked you(Skinny Pete) to solve it for him. You have to answer for T testcases.

Input
First line contains T number of testcases.

The only line of each testcase will contain k (desired sum/code in Walts notes).

$1 \leq T \leq 10^5$

$1 \leq k \leq 10^{18}$

Output
You have to print output(minimum number of elements to be used) for each testcase in new line.

Example

input Copy

```
3
2
5
7
```

output Copy

```
1
2
3
```

Note
The answers can be explained as follows:-

$2 = 2^1$. Only 1 number is used.

$5 = 2^2 + 2^1$. Only 2 numbers are used.

$7 = 2^2 + 2^1 + 2^0$. Only 3 numbers are used.

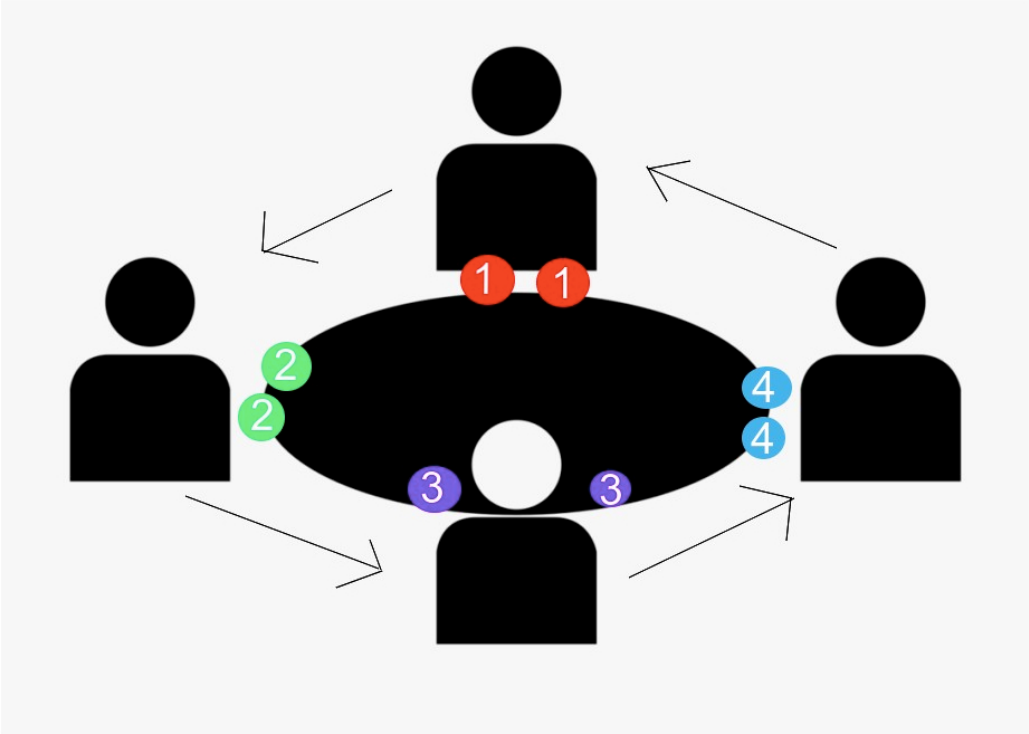
B. Rotating Balls

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

N persons are sitting on a round table. Being bored they decided to play a game in which each of them were given two balls with the same number written on both the balls. For example, the first person was given two balls with number 1, second person with number 2, and N^{th} person with number N. In one move each of them will pass the ball with a smaller number with him/her to the person sitting right (In case the numbers on the balls are same, the person can pass any ball).

You need to find if it is possible that anytime after at least one move, at least one person is having both the balls of the same number (It might be different from the initial ball he was assigned).

Initial arrangement is shown below.



Input
First Line contains one Interger T, the number of test cases.

For each of the next T lines, it contains one Interger N.

$1 \leq T \leq 10^5$

$1 \leq N \leq 10^9$

Output
Print Yes if it is possible else print No.

You may print every letter in any case you want.

Example

input	Copy
2 1 3	
output	**Copy**
Yes Yes	

Note
The output is self-explanatory. Please simulate the given task yourself.

C. Minimized Way

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

Jonas wants to meet Martha. There is a river in Winden(the city where Martha and Jonas live). Jonas has to cross the river. There are N rocks in the river. Each rock is given one number denoted by $A[i]$. Jonas can jump from i^{th} rock to the smallest j^{th} rock if $i < j \leq N$ and either of conditions satisfies.

1. $A[j] \geq A[i]$ and there is no k such that $A[k] \geq A[i], where\ i < k < j$

2. $A[j] < A[i]$ and there is no k such that $A[k] < A[i], where\ i < k < j$

Now each rock absorbs some energy from Jonas's body(Winden is a magical city don't think about reality). The i^{th} rock absorbs $val[i]$ energy from Jonas's body.

You have to find the minimum energy Jonas has to use to reach N^{th} rock.

Initially, Jonas is on 1^{st} rock.

Input
The first line contains T - the number of test cases.($1 \leq T \leq 10$)

For each test case, the first line contains N - the number of rocks in the river.($1 \leq N \leq 10^5$)

For each test case, the second line contains N space-separated integers $A[i]$. For $1 \leq i \leq N$ ($1 \leq A[i] \leq 10^5$)

For each test case, the third line contains N space-separated integers $val[i]$. For $1 \leq i \leq N$ ($1 \leq val[i] \leq 10^5$)

Output
For each testcase, print the answer in a new line.

Example

input Copy

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

output Copy

```
11
```

Note

The path for the testcase is rock numbers 1->2->4->5

D. Grand Line

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

In the Grand Line Ocean, there are dozens and dozens of Islands. Luffy and StrawHat Pirates started from the first Island of Grand Line and started visiting every other Island.

To our astonishment, Luffy conquered the Grand Line and made a map before he retired as a Pirate.

The map contains islands, named from **Island 1** to **Island N**.

For any two islands **Island I** and **Island J** ($I \neq J$), you can go to **Island J** from **Island I** only if the value **I** divides **J**. (**I** is a divisor of **J**)

Given the the value for the last island, **Island N**, equal to **N**, find the number of ways to reach **Island N** from the First Island, that is, **Island 1**.

Note that to reach **Island 1** from **Island 1**, you don't need any way(s) because you are already there.

Input

The input consists of multiple test cases. The first line contains an integer T $(1 \leq T \leq 10^5)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer N $(1 \leq N \leq 10^6)$ — the Island to reach, as described in the problem statement.

Output

For each test case , print an integer denoting the number of ways to reach **Island N** from **Island 1**.

Example

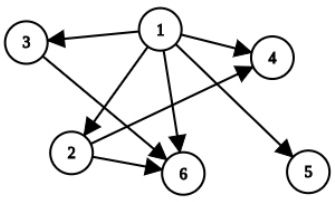
input Copy

```
3
2
4
6
```

output Copy

```
1
2
3
```

Note



The above graph will suffice for all the 3 test cases. As seen in the graph,
There exists only 1 way to reach Island 2 from Island 1. (1 — 2)

There exists 2 ways to reach Island 4 from Island 1. (1 — 4 | 1 — 2 — 4)

There exists 3 ways to reach Island 6 from Island 1. (1 — 6 | 1 — 2 — 6 | 1 — 3 — 6)

E. Too much DARK

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

There are **N** worlds connected through "Time-travel". Two worlds A, B are called ***adjacent*** through "Time-travel" if a person can go from world-A to world-B and world-B to world-A using a Time machine.

Luckily all **N** world are connected through "Time-travel" such that it forms a **Tree**. Recall that a tree on **N** vertices is an undirected, connected graph with **N−1** edges.

Recently Martha found *"The Portable Time Machine"* in the *Winden Caves*. She is very excited and wants to travel through time. she can either go to any ***adjacent world*** or stay in the same world. Jonas has a *"Golden Time Travel Sphere"*. Using it, he can also go to the ***adjacent world*** or stay in the same world

They travel **alternatively**. Martha travel first, then Jonas, then Martha, so on...

As *"Golden Time Travel Sphere"* is made from *"God Particle"*, excessive use of it can trigger Apocalypse. So Jonas wants to know in *how many moves* he would **surely** meet Martha.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer N ($1 \leq N \leq 10^5$) — the number of worlds.

Each of the next $N-1$ lines contains two integers u, v ($1 \leq u, v \leq N$) — It means, world u, v are ***adjacent***.

next line contains two integers X, Y ($1 \leq X, Y \leq N$) — the initial position of Jonas and Martha respectively.

It's guaranteed that the given graph is a tree.

It's guaranteed that the sum of N for all test cases does not exceed 10^5.

Output

For each test case , print an integer denoting **minimum** moves required by Jonas to ensure that he definitely meets Martha, seperated by new line. (Remember, Martha may choose to move away from Jonas or towards Jonas. You need to print the minimum moves required by Jonas to ensure that he will meet Martha in any case)

Examples

input Copy

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

output Copy

```
5
```

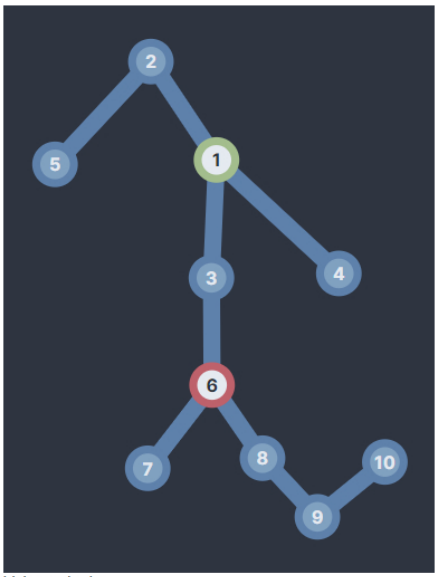
input Copy

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

output Copy

```
0
```

Note



let J=world in which Jonas is, M = world in which Martha is

initially $\rightarrow$ J=1, M=6

after move of Martha $\rightarrow$ J = 1 , M = 8

after move of Jonas $\rightarrow$ J = 3 , M = 8 (Count = 1)

after move of Martha $\rightarrow$ J = 3 , M = 9

after move of Jonas $\rightarrow$ J = 6 , M = 9 (Count = 2)

after move of Martha $\rightarrow$ J = 6 , M = 10

after move of Jonas $\rightarrow$ J = 8 , M = 10 (Count = 3)

after move of Martha $\rightarrow$ J = 8 , M = 10

after move of Jonas $\rightarrow$ J = 9 , M = 10 (Count = 4)

after move of Martha $\rightarrow$ J = 9 , M = 10

after move of Jonas $\rightarrow$ J = 10, M = 10 (Count = 5)

They meet in world 10 by using *"Golden Time Travel Sphere"* 5 times.

It can be shown that Jonas will catch Martha in 5 moves no matter where Martha goes.

In second testset they are already in same world.

F. AI Apocalypse

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

This is an interactive problem.

This is Age 2071.

Humans are on the verge of extinction. Only the ones destined to survive are living, their lives full of despair. They fear their own creation, the AI technology developed by them to fight the COVID19 Virus in Age 2020, named AliceAI.

Only the chosen humans, the best among the survivors are given the task to capture and deactivate AliceAI.

AliceAI, an AI based cyborg, roams around the human world. For the AI, the human world comprises of a single line with **N** positions from **1** to **N**.

AliceAI roams continously from position **1** to position **N**, both inclusive.

Note that it can not move in any other positions besides those mentioned in the previous statement.

The cryptography and reverse engineering experts from the survivors have decoded AliceAI's pattern of movement which is as follows:-

$$F(X)=\begin{cases} X+1 & if\ X=1 \\ X-1 & if\ X=N \\ (I,J)\ such\ that\ I>X & if\ 1<X<N \\ and\ J<X\ and\ I\times J=X^2-1 & \\ and\ |(I-J)|\ is\ minimum & \end{cases}$$

In the above Function,

- X is the current position of AliceAI
- $1 \leq X \leq N$
- When $F(X) = (I, J)$,where I and J satisfy all the conditions, AliceAI can choose to either go to position I or position J, but not both.

Your mission is to find the correct position of AliceAI (intercept AliceAI), so that your team can deactivate it and save the remaining world from the catastrophe.

Your program can ask at most $(2 \times N)$ - 2 queries.

Each query must follow the format given below:-

- $A\ a$: Checks whether AliceAI is currently present at a. Prints Yes if it is present at a, othewise No.

Only after you ask the query, AliceAI uses the above stated Function $F(x)$ to move to it's next location from it's current location, x.

Once you receive Yes, you need to terminate your Program immediately.

Input

The first line contains one integer N ($2 \leq N \leq 2 \times 10^4$). The remaining parts of the input will be given throughout the interaction process.

Interaction

In each query, your program needs to print a line that contains one uppercase letter A and an integer $a(1 \leq a \leq N)$.

If your query receives Yes, then you need to terminate your Program immediately.

After providing each line in the output, don't forget to flush the output. To do it use:

- fflush(stdout) in C/C++ (Using endl in C++ fulfills the same purpose);
- System.out.flush() in Java;
- sys.stdout.flush() in Python;
- flush(output) in Pascal;

See the documentation for other languages.

Example

input Copy

```
4
No
No
No
Yes
```

output Copy

```
A 1
A 4
A 1
A 3
```

Note

The explanation of the example is as follows:-

The value of N given is 4.

Now, the program asks the query to check whether AliceAI is at position 1. (First line of your output)

The interactor returns No.

Then the program asks the query with $a = 4$.

The interactor again returns No.

Similarly the program checks for $a = 1$ and $a = 3$.

The interactor returns No and Yes respectively.

As soon as the interactor returned Yes, the program terminated.