

Ada Invents Programming

Problem ID: adalovelace



Augusta Ada King, Countess of Lovelace (10 December 1815 – 27 November 1852), was an English mathematician and writer, chiefly known for her work on Charles Babbage's proposed mechanical general-purpose computer, the Analytical Engine. She was the first to recognize that the machine had applications beyond pure calculation, and to have published the first algorithm intended to be carried out by such a machine. As a result, she is often regarded as the first computer programmer.¹

Our story happens on one of those programming days for Ada. She plans to write a program that requires n units of programming. Ada also modeled her current level of programming skill by a positive integer a . That means that in a day, she can complete any program that is less than or equal to a units of programming or, for any program that needs more than a units of programming, reduce the remaining amount by a . Furthermore, Ada thinks that her programming skills will improve over time. After every b days of programming, her programming skill level goes up by one. So on the day $b + 1$, she will be able to complete $a + 1$ units of programming, and on the day $2b + 1$, she will be able to complete $a + 2$ units of programming, and so on.

Ada wants to know how many days she needs to finish the program. But instead of solving it herself, she gave this task to you so that your programming skill will level up in no time!

Input

The input contains a single line of 3 positive integers ($1 \leq n \leq 100000$), ($1 \leq a \leq 1000$), ($1 \leq b \leq 100$), separated by space.

Output

Output a single line with a single integer, the number of days Ada needs to finish the program.

Sample Input 1

55 1 1

Sample Output 1

10

Sample Input 2

100 10 7

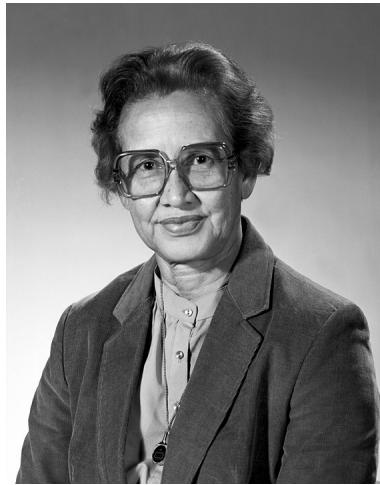
Sample Output 2

10

¹Wikipedia contributors. (2023, February 8). Ada Lovelace. Wikipedia. https://en.wikipedia.org/wiki/Ada_Lovelace

Katherine Computes the Universe

Problem ID: katherinejohnson



Katherine Johnson (born Coleman; August 26, 1918 – February 24, 2020) was an American mathematician whose calculations of orbital mechanics as a NASA employee were critical to the success of the first and subsequent U.S. crewed spaceflights. During her 33-year career at NASA and its predecessor, she earned a reputation for mastering complex manual calculations and helped pioneer the use of computers to perform the tasks. The space agency noted her "historical role as one of the first African-American women to work as a NASA scientist".¹

One day, Katherine was presented with this fascinating sequence which is said to contain the secret of the universe!

42, 4211, 421112, 42121131, 42121311111211, 42121312111111512, 42121312151211111117111, ...

The sequence starts with 42, and then each following member “reads” the previous member. For instance, the second member, 4211, says the previous member is “a block of 4, a block of 2, the first block has a length of 1, and the second block has a length of 1”, and the third member, 421112 says the previous member is “a block of 4, a block of 2, a block of 1, the first block has a length of 1, the second block has a length of 1, the third the block has a length of 2”, and so on.

Help our heroine unravel the mystery by computing the n -th member of the sequence!

Input

The input contains a single line of a positive integer ($1 \leq n \leq 32$).

Output

Output a line with a single number: the n -th member of the sequence.

Sample Input 1	Sample Output 1
1	42
Sample Input 2	Sample Output 2
4	42121131
Sample Input 3	Sample Output 3
9	421213121512171813111111111111111113111

¹Wikipedia contributors. (2023b, February 14). Katherine Johnson. Wikipedia. https://en.wikipedia.org/wiki/Katherine_Johnson

Time Smiles

Problem ID: timesmiles

(...)
*we could make time smile
when we danced*
(...)

Badlands I
Caio Dias Valentim (1987 – 2022)

Time smiles every time Josephine¹ and Maria² dance (and only when they dance). This means that whenever Josephine and Maria dance time has the shape below.

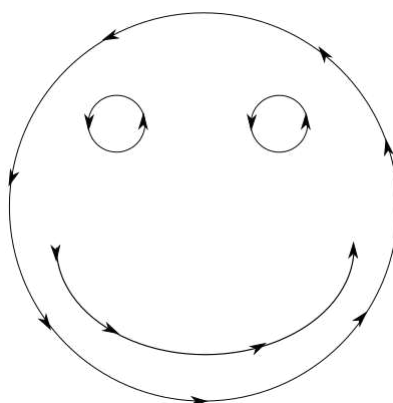


Figure 1: Time whenever Josephine and Maria dance

In other words, time becomes **exactly three loops and one segment**.

While a smiling time creates problems for historians and linguists (which tense to use when something is in the past and in the future at the same time?), philosophers think that it is a very elegant solution to the age old question of “which came first, the chicken or the egg?” The answer being “Josephine and Maria were dancing, so the chicken and the egg appeared before each other on time’s left eye”.

In this problem you are asked to verify if Josephine and Maria are dancing or not. For that you observe N events, and for each event i you know which event a_i happened after i , but before any other event. For example, suppose that you are given the following information (where $A \rightarrow B$ means B follows A , and none of the other events in the list happens between A and B , and \perp means “nothing”. In the example below we don’t observe any event after 5):

1 \rightarrow 3 2 \rightarrow 5 3 \rightarrow 1 4 \rightarrow 2 5 \rightarrow \perp 6 \rightarrow 9 7 \rightarrow 7 8 \rightarrow 6 9 \rightarrow 8

In the example above Josephine and Maria are dancing, since time has 3 cycles (1 \rightarrow 3 \rightarrow 1, 7 \rightarrow 7, and 6 \rightarrow 9 \rightarrow 8 \rightarrow 6) and one segment (4 \rightarrow 2 \rightarrow 5). All components of time (3 circles and 1 segment) should contain at least one element.

Input

The first line of input will contain a single integer N ($1 \leq N \leq 10^5$) with the numbers of events observed. Events are numbered from 1 to N .

¹<https://www.womenshistory.org/education-resources/biographies/josephine-baker>

²<https://www.womenshistory.org/education-resources/biographies/maria-tallchief>

The second line will contain N integers representing the successor of the events in order. If an element of this line is -1 this indicates that the corresponding event has no successor. All integers in this line will be either -1 or be between 1 and N .

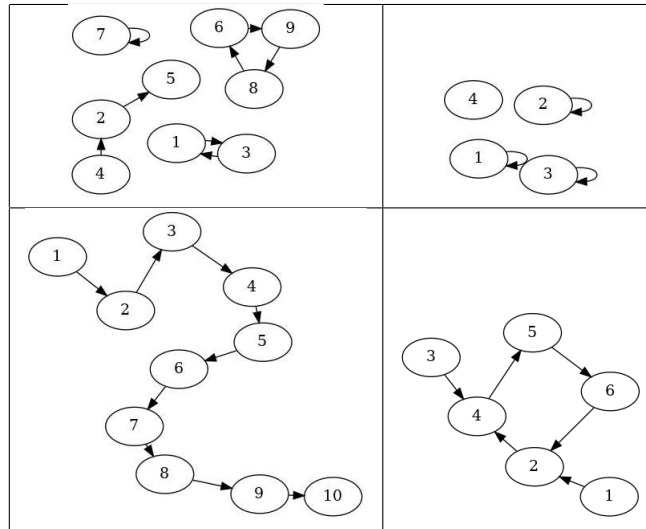


Figure 2: Time described in the four samples below

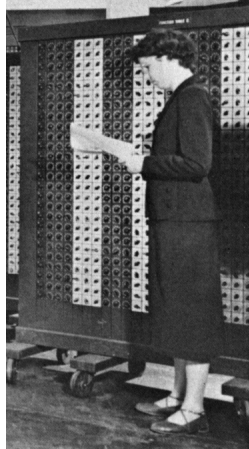
Output

The output should contain a single character. “Y” if Josephine and Maria are dancing and “N” if they aren’t.

Sample Input 1	Sample Output 1
9 3 5 1 2 -1 9 7 6 8	Y
Sample Input 2	Sample Output 2
4 1 2 3 -1	Y
Sample Input 3	Sample Output 3
10 2 3 4 5 6 7 8 9 10 -1	N
Sample Input 4	Sample Output 4
6 2 4 4 5 6 2	N

Betty Raises a Point

Problem ID: bettyholberton



Frances Elizabeth Holberton (born Snyder; March 7, 1917 – December 8, 2001), also known as Betty Holberton, was an American computer scientist who was one of the six original programmers of the first general-purpose electronic digital computer, ENIAC. She also developed the C-10 instruction set for BINAC, considered the prototype of all modern programming languages. She participated in developing early standards for the COBOL and FORTRAN programming languages together with Grace Hopper. She also invented breakpoints in computer debugging.¹

Keyboard shortcuts are everywhere in computer user interfaces nowadays. For instance, “ctrl+c” is often used for copying, “ctrl+v” for pasting, and so on. To press a shortcut, one must press all the keys in the shortcut. However, things were much more difficult in the old days, as a user could not press multiple keys all at once but only one more key at a time.

Betty found a potential design flaw in a shortcut system: A user may unwittingly trigger another shortcut before pressing all the keys they need for the intended shortcut. For example, a user might want to press the shortcut “a+b+c” while “a+b” may be a shortcut for something else. If the user first presses “a” then “b”, and is going to press “c”, they will trigger the “a+b” shortcut first by accident! Worse, some shortcuts may be impossible because some other shortcuts will always be triggered first, no matter the order of the keys pressed.

Help Betty fix the design by identifying which shortcuts are problematic. You are given a list of shortcuts in the system and decide which one of the three following cases each shortcut belongs.

1. “WARNING”: A user may trigger some other shortcut first, but this can be prevented by pressing the keys in a specific order.
2. “ERROR”: A user will always trigger some other shortcut first, no matter in which order the keys are pressed.
3. “OK”: A user can always trigger this shortcut successfully, no matter in which order the keys are pressed.

Input

The first line contains a single integer $1 \leq n \leq 2000$, the number of shortcuts in the list.

The following n lines each contain a string of lowercase letters and digits of no more than 10 characters, describing a shortcut.

It is guaranteed that each shortcut is unique.

Output

Output n lines. Each line is either “WARNING”, “ERROR”, or “OK”, the verdict for the i -th shortcut in the input.

¹Wikipedia contributors. (2023, January 6). Betty Holberton. Wikipedia. https://en.wikipedia.org/wiki/Betty_Holberton

Sample Input 1

5
ab
bc
ca
abc
abcd

Sample Output 1

OK
OK
OK
ERROR
WARNING

Sample Input 2

3
1
23
321

Sample Output 2

OK
OK
WARNING

Thelma Liberates the Coders

Problem ID: thelmaestrin



Thelma Estrin (born Austern; February 21, 1924 – February 15, 2014) was an American computer scientist and engineer who did pioneering work in the fields of expert systems and biomedical engineering. Estrin was one of the first to apply computer technology to healthcare and medical research. She was also passionate about women's studies. In a 1996 paper, she connected feminist epistemology and its pedagogical values to how computer science could become “more relevant for minority and low-income students.”¹

The legend tells us of an island of coders that was discovered by Thelma. Everyone on coder island has a unique id and loves coding. What is peculiar about this island is that it follows a particular scheduling process to decide who should contribute how many lines of code for a new project, as described below:

- Step 0: the coders use scientific methods to determine that the new project needs m lines of code in total.
- Step 1: the coder with the lowest id remaining proposes a distribution of work, i.e., how many lines of code each coder should contribute to the project.
- Step 2: all remaining coders, including the proposer, vote on whether to accept this proposal. In case of a tie, the proposer has the casting vote.
- Step 3: If the majority passes the proposal, then all the remaining coders will work on the project, and the m lines of code are distributed as proposed to each coder; otherwise, the proposer is forced to take a vacation, so they leave the scheduling meeting and may not contribute to the project. The remaining coders continue with Step 1.

You may wonder why this system would work in any sensible way. This is because the habitants of coder island hold coding so dearly in their hearts, and they find vacation much less exciting than working on projects and solving challenging problems. Each coder, when proposing or voting for a proposal, always prioritizes the following in the listed order:

1. Avoid taking a vacation. So they could at least work on the project, although in some cases, they may not be able to contribute any code.
2. Maximize the number of lines of code they themselves contribute.
3. If multiple proposals would give the same results for the first two priorities, they prefer to have more coders on vacation to keep the dev team small and focused.

¹Wikipedia contributors. (2023, February 16). Thelma Estrin. Wikipedia. https://en.wikipedia.org/wiki/Thelma_Estrin

4. If there is still a tie, they will then try to maximize the number of lines of code for the coder with the largest id to contribute, favoring giving the newer coder more opportunities and then maximizing the number of lines for the second-largest, and so on.

The scheduling meetings of the coders often go very long and tedious until Thelma gives them the project “liberator”: write a program that takes in the number of coders and the lines of code needed and outputs the final result following the process above.

Input

The input contains a single line of two integers ($1 \leq n \leq 1000$) the number of coders on the island, and ($1 \leq m \leq 100000$) the total lines of code to be distributed among the coders.

It is **not** guaranteed that the number of coders is less than the total lines of code needed.

Output

Output n lines, the i -th line describes the coder with i -th lowest id. Output -1 if the i -th coder will be forced to take a vacation. Otherwise, output a non-negative integer, the number of lines of code the i -th coder needs to contribute to the project.

Sample Input 1

```
5 100
```

Sample Output 1

```
98
0
1
0
1
```