Problem G
Group Testing

During the first outbreak of the COVID-19 pandemic, extensive population-wide testing proved to be one of the best strategy to control and prevent the outbreak. However, PCR testing costs both time and money. An effective way to speed up the process and to save our resources is to implement group testing (or pool testing) strategy.

In this method, instead of doing test for individuals, we mix the samples of p individuals and run the test. If the result is negative, we can safely conclude that all p individuals are negative. Otherwise, if the result is positive, individual testing is used to determine who has the virus.

Today, we have a line of n people coming back from foreign countries. They are numbered 1 to n in the order they are lining up to do the testing before immigrate to Vietnam. Person 1 is in the front of the line, an person n is in the back of the line.

Historical data shows that these people have very little chance of getting the virus because they had already tested negative before their flights. While arriving to Vietnam, there should be at most 2 people with the virus. And if there are exactly two people with the virus, they must be in *consecutive* positions.

The challenge is that we only have 10 test kits left. Each test takes hours to finish so we want to do 10 tests in parallel. We need to determine whether there are people with the virus or not; and if there are at least one people with the virus, we need to determine potision of exactly one person with the virus.

More formally, you are about to write a program which reads the number of people n, then provides a strategy to do 10 tests in parallel. After that, your program receives the outcome of all 10 tests and concludes the result as below:

- If no people have the virus, your program should conclude that there is no positive case.
- If exactly one person at position z has the virus, your program should find out this position z.
- If exaclty two people at positions z and $z + 1$ have the virus, your program should find out either position z or $z + 1$. Any of these value will be accepted.

Interaction protocol

- Firstly, your program reads the integer n — the number of people ($1 \le n \le 1000$).
- Secondly, your program prints 10 lines, each line represents a test you would like to do in the following format:
  - Each line starts with an integer p ($0 \le p \le n$) — the number of people involved in this test.
  - Then the line follows by p distinct integers $x_1, x_2, \ldots, x_p$ ($1 \le x_i \le n$) — the indices of the people involved in this test.
- Thirdly, your prgram reads 10 lines representing the outcome of all 10 tests. Each line is either `POSITIVE` or `NEGATIVE`. `NEGATIVE` means that everyone involved in the test

does not have the virus; while `POSITIVE` means that **at least one** involved person has the virus, but we do not know how many people or which exact people have the virus.

Please be aware that you can not read any of these lines from the standard input before having printed exactly 10 **lines in the previous step.**

- Finally, your program concludes the result by print out a single integer:
  - If no people have the virus, print -1.
  - Otherwise, print z ($1 \leq z \leq n$) meaning that the person at position z has the virus.
- After that, your program should terminate with exit code 0.

Sample communication

standard input	standard output
10	

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

NEGATIVE NEGATIVE NEGATIVE NEGATIVE NEGATIVE POSITIVE POSITIVE POSITIVE NEGATIVE NEGATIVE	
	7

Note

When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the interactor. In order to avoid such situation **you have to use special 'flush' operation each time you output some data**. There are these 'flush' operations in standard libraries of almost all languages. For example, in C++ you may use 'fflush(stdout)' or 'cout « flush' (it depends on what do you use for output data — 'scanf/printf' or 'cout'). In Java you can use method 'flush' for output stream, for example, 'System.out.flush()'. In Python you can use 'stdout.flush()'.