Dragon Ball

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

This is an interactive problem

Again, as in any other battle where he was involved, Krilin has been killed! And as always, Goku has to resurrect him using dragon balls...

Thanks to the dragon radar created by Bulma, Goku was able to locate some dragon balls. The device indicates that they are scattered somewhere in the taxiway of Satan City airport.

The airport can be represented as an infinite 2D grid. Goku is using a push-back vehicle (the vehicle used to move planes) in order to move in the airport. He is now at the cell located in $(0, 0)$ and can only move to an adjacent cell in one of the four directions: 'N' (North),'S' (South),'E' (east) and 'W' (West). We will use a string of these characters to describe his path, hence 'SEEN' means he first moved one cell to the south, two cells to the east and one cell to the north.

Goku is seeking your help, he wants you to determine the path to the dragon ball. You can send queries to the dragon radar, each query is a string consisting of the letters {'N','S','E','W'} describing a path. The dragon radar will return an integer equals to the length of the longest substring of the query, that is also a prefix of the **unique** path to the dragon ball. It is obvious that the path to the dragon ball will not contain two consecutive moves of opposite directions (North then south for instance).

Input

The first line contains an integer n $(1 \leq n \leq 1000)$ — the length of the path to the dragon ball.

Following lines will contain responses to your queries. The i-th line is a response to your i-th query. This line will contain an integer x denoting the length of the largest substring of your query that is a prefix of the path to the dragon ball.

The testing system will allow you to read the response on the query only after your program prints the query for the system and performs a `flush` operation.

Output

To make the queries your program must use standard output.

Your program must print at most $n+5$ queries — Strings x_i of size at most 10^4 consisting of the characters {'N','S','E','W'}, one query per line (do not forget "*end of line*" at the end of each query).
After printing each line your program must perform operation `flush`.

When your program will guess the path print "`S ?`", where S is the answer and terminate your program.

Interaction Protocol

To make a query to the dragon radar, print a string S ($1 \leq |S| \leq 10^4$), containing only the characters described above.

After each query, read one integer x that corresponds to the length of the longest substring of S that is a prefix of the desired path. If $x = -1$, it means that you asked more queries than allowed or asked an invalid query. Your program should immediately terminate (for example, by calling `exit(0)`). You will receive `"Wrong Answer"`if you asked more queries than allowed or asked an invalid query. If you ignore this, you can get other verdicts since your program will continue to read from a closed stream.

Your solution will get `"Idleness Limit Exceeded"`, if you don't print anything or forget to `flush` the output, including the final answer.

To `flush` you can use (just after printing a query and line end):

— `fflush(stdout)` in C++ ;

— `System.out.flush()` in JAVA;

— `flush(output)` in Pascal;

— `sys.stdout.flush()` in Python.

Example

standard input	standard output
4	S
1	SE
2	SEE
3	SEEN
4	SEEN ?