This is a communication (interactive) task!

Hidden Sequence

You did it! You have finally made it to the X spot of the treasure map, but, to open the treasure chest, you first need to break the code. On the label attached to the chest, the rules of the game are written: the code is a binary sequence of length N, and in order to find this sequence you're allowed to ask questions of the following type: "is S a subsequence of the hidden sequence?"

A binary sequence S, with values $S_1, S_2, \dots, S_K$ is considered to be a subsequence of the code C, with values $C_1, C_2, \dots, C_N$, if and only if S can be obtained by deleting some of the values of C.

Formally, S is a subsequence of C if and only if there exist K values $1 \leq i_1 < i_2 < \cdots < i_K \leq N$ such that $C_{i_j} = S_j$ for any $1 \leq j \leq N$.

You will get a part of the treasure if you succeed in cracking the code. Depending on how short the longest query you've asked for is, you'll get more or less of the treasure (which, in our case, is represented by 100 points)

Implementation details:

You should include the library "grader.h" at the beginning of your source code and implement the following procedure:

```
vector < int > findSequence (int N);
```

- The procedure is called exactly once by the grader
- N is the length of the hidden sequence
- The function should return the hidden sequence as a STL vector of ints containing 0s and 1s

Your code may contain any number of additional procedures and may declare global variables.

Your solution can call the following function any number of times, as long as the overall process fits in the time limit:

```
bool isSubsequence (vector < int > S);
```

- S is a STL vector of ints that should consist only of 0s and 1s that you choose
- The procedure returns 1 if S is a subsequence of the hidden code, and 0 otherwise

Subtasks and scoring

Let L be the length of the longest asked query. There are 2 subtasks for the problem. For each subtask you will receive a score equal to the smallest score obtained on one of the tests in that subtask:

Subtask	Restriction	L condition	Score
1	$7 \leq N \leq 10$	$L \leq \left\lfloor \frac{N}{2} \right\rfloor + 1$	20
		$L \leq \left\lfloor \frac{3N}{4} \right\rfloor + 1$	15
		$L \leq N$	10
2	$100 \leq N \leq 200$	$L \leq \left\lfloor \frac{N}{2} \right\rfloor + 1$	80
		$L \leq \left\lfloor \frac{N}{2} \right\rfloor + 3$	72
		$L \leq \left\lfloor \frac{3N}{4} \right\rfloor + 1$	44
		$L \leq N$	24

We've noted by $\lfloor X \rfloor$ the integer part of X. As you can see, if any of the queries you've asked had a length greater than N, you will get 0 points.

Example of interaction

The grader calls the function findSequence (3). The hidden sequence is 0 1 0. Your source calls the functions isSubsequence ([0, 1]) and isSubsequence ([1, 0]) and receives a positive answer for both queries. It then calls isSubsequence ([1, 1]), which returns a negative answer (0) and your program returns the sequence [0, 1, 0] by using the information it has received.

Sample grader

You're provided with an archive containing a sample grader to help you test your sources locally. It includes a header file, a file grader.cpp, and a file sequence.cpp that you need to implement.

The grader reads from standard input the hidden sequence and then calls the function you've implemented to find the sequence. In the end, it checks if the returned sequence matched the hidden one and, if so, it prints the maximum length of a query you've asked for to the standard output.

The format of the input the grader reads is:
N
$C_1\ C_2\ C_3\ \ldots\ C_N$

The sample grader is **not** the one used for evaluation, but one that should assist you in your local testing. Of course, you are allowed to change it, but the source you submit should still respect the format (by including "grader.h" and implementing the function findSequence)