CONSUL

Maximum time of execution: 1 seconds/test.
Maximum available memory: 64 MB

In Reme, a consular election is being held. This election has N electors, each of which can vote with one of the 1.000.000.000 different candidates. A candidate is considered a winner if she wins strictly more than one third of the vote (this happens because Reme elects two consuls each year). Now, you are an aspiring vote-rigger. In order to more effectively rig the vote, you need to know at least one prospective winner. Unfortunately, although each elector has cast their vote, it is not public — at least not without a price. Moreover, the old consuls, who have already counted votes, can tell you the number of votes a particular candidate won — again, at a price.

More formally, you are asked to consider a hidden sequence v of N integers v[1] … v[N], with values at least 0 and less than 1.000.000.000. You can perform two types of queries on this sequence:

- you can find out the value of a particular element v[i] in the sequence
- you can see how many times a particular value x appears in the sequence.

You are asked to find a value x that appears strictly more than N/3 times in the sequence, while making the number of queries you perform small enough.

TASK

Given N, find out any winner of the election, using the given queries, or announce that no winner exists.

INTERACTION

This is an interactive task. Thus the contestant will interact with the following functions, after including the file `grader.h`. The contestant must implement a function `void solve(int N),` where the parameter represents the N from the problem statement, which must solve an instance of the task **(note that in one run of the source code, this function may be called multiple times)**.In order to solve the task, the contestant may use the following functions:

- `int kth(int i)` — which will return the value of v[i]
- `int cnt(int x)` — which will return the frequency of `x` in the array v
- `void say_answer(int a)` — which, if `a` is at least 0 and less than 1.000.000.000, signals that `a` is a winner of the election; and if `a` is -1, that no winner exists. If there exists a winner, and `a` is not equal to a winner of the election, or if there is no winner, but `a` is not -1, or `a` is neither a valid candidate nor -1, then the contestant's solution will get verdict "Wrong Answer!".

CONSTRAINTS

Subtask	Score	Restrictions
1	15 points	N <= 50
2	another 20 points	N <= 100
3	another 65 points	N <= 1000

SCORING:

Let Q be the maximum number of queries made in any test case in a file (not including the call of say_answer).

Subtask 1:

- **If Q <= 50, then you will receive 100% of the points on the test case.**
- **If Q > 50, then you will receive 0% of the points on the test case.**

Subtask 2:

- **If Q <= 60, then you will receive 100% of the points on the test case.**
- **If 60 < Q <= 110, then you will receive (220-2Q)%of the points on the test case.**
- **If Q > 100, then you will receive 0% of the points on the test case.**

Subtask 3:

- **If Q <= 60, then you will receive 100% of the points on the test case.**
- **If 60 < Q then you will receive (0.9^(Q - 60) * 100)%of the points on the test case.**

NOTE: the number of test cases is at most 16000