Day 1 Task 2: Hotter Colder

Jack and Jill play a game called *Hotter, Colder*. Jill has a number between 1 and N, and Jack makes repeated attempts to guess it.

Each of Jack's guesses is a number between 1 and N. In response to each guess, Jill answers *hotter*, *colder* or *same*. For Jack's first guess, Jill answers *same*. For the remaining guesses Jill answers:

- *hotter* if this guess is closer to Jill's number than his previous guess
- *colder* if this guess is farther from Jill's number than his previous guess
- *same* if this guess is neither closer to nor further from Jill's number than his previous guess.

You are to implement a procedure **HC(N)** that plays Jack's role. This implementation may repeatedly call **Guess(G)**, with **G** a number between 1 and N. **Guess(G)** will return 1 to indicate *hotter*, -1 to indicate *colder* or 0 to indicate *same*. **HC(N)** must return Jill's number.

Example

As example, assume N=5, and Jill has chosen the number 2. When procedure **HC** makes the following sequence of calls to **Guess**, the results in the second column will be returned.

Call	Returned value	Explanation
Guess(5)	0	Same (first call)
Guess(3)	1	Hotter
Guess(4)	-1	Colder
Guess(1)	1	Hotter
Guess(3)	0	Same

At this point Jack knows the answer, and **HC** should return 2. It has taken Jack 5 guesses to determine Jill's number. You can do better.

Subtask 1 [25 points]

HC(N) must call **Guess(G)** at most 500 times. There will be at most 125 250 calls to **HC(N)**, with N between 1 and 500.

Subtask 2 [25 points]

HC(N) must call **Guess(G)** at most **18** times. There will be at most 125 250 calls to **HC(N)** with N between 1 and 500.

Subtask 3 [25 points]

HC(N) must call **Guess(G)** at most **16** times. There will be at most 125 250 calls to **HC(N)** with N between 1 and 500.

Subtask 4 [up to 25 points]

Let W be the largest integer, such that $2^W \leq 3N$. For this subtask your solution will score:

- 0 points, if **HC(N)** calls **Guess(G)** 2W times or more,
- 25α points, where α is the largest real number, such that $0<\alpha<1$ and **HC(N)** calls **Guess(G)** at most 2W-αW times,
- 25 points, if **HC(N)** calls **Guess(G)** at most W times.

There will be at most 1 000 000 calls to **HC(N)** with N between 1 and 500 000 000.

*Be sure to initialize any variables used by **HC** every time it is called.*

Implementation Details

- Use the RunC programming and test environment
- Implementation folder: `/home/ioi2010-contestant/hottercolder/` (prototype: hottercolder.zip)
- To be implemented by contestant: `hottercolder.c` or `hottercolder.cpp` or `hottercolder.pas`
- Contestant interface: `hottercolder.h` or `hottercolder.pas`
- Grader interface: `grader.h` or `graderlib.pas`
- Sample grader: `grader.c` or `grader.cpp` or `grader.pas` *and* `graderlib.pas`
- Sample grader input: `grader.in.1` `grader.in.2`.
  Note: The input file contains several lines, each containing N and Jill's number.
- Expected output for sample grader input: the grader will create files `grader.out.1` `grader.out.2` etc.
  - If the implementation correctly implements Subtask 1, one line of output will contain `OK 1`
  - If the implementation correctly implements Subtask 2, one line of output will contain `OK 2`
  - If the implementation correctly implements Subtask 3, one line of output will contain `OK 3`
  - If the implementation correctly implements Subtask 4, one line of output will contain `OK 4 alpha α`
- Compile and run (command line): `runc grader.c` or `runc grader.cpp` or `runc grader.pas`
- Compile and run (gedit plugin): *Control-R*, while editing any implementation file.
- Submit (command line): `submit grader.c` or `submit grader.cpp` or `submit grader.pas`
- Submit (gedit plugin): *Control-J*, while editing any implementation or grader file.