Demonstration Task 1: Guess

Jill is thinking of a number between 1 and N, and Jack wants to guess it by asking Jill questions of the form "Is it bigger than K?" for K between 1 and N.

You are to implement a procedure **play(N)** that implements Jack's role in the game. Your implementation should repeatedly call the procedure **bigger(K)**, which is implemented by the grader. **bigger(K)** will return 1 if Jill's number is greater than K; otherwise it will return 0. Jill's number should be returned by your implementation as the result of **play**.

Subtask 1 [50 points]

Assume that N=16. Your implementation must use at most 15 calls to **bigger** and must return the correct result. *The implementation files described below contain a correct implementation of this subtask.*

Subtask 2 [50 points]

Assume that N=16. Your implementation must use at most 4 calls to **bigger** and must return the correct result.

Implementation Details

- Use the RunC programming and test environment
- Implementation folder: `/home/ioi2010-contestant/guess/` (download prototype here)
- To be implemented by contestant: `player.c` or `player.cpp` or `player.pas`
- Contestant interface: `player.h` or `player.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 sample grader reads N and Jill's number from standard input.
- Expected output for sample grader input: `grader.expect.1` `grader.expect.2`
- 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.*
- *CPU time limit: 10 seconds*
- *Memory limit: 256 MB*