

Guess the Number

The grader has chosen a hidden integer H between 1 and N .

Your task is to find H by asking comparison questions.

Implementation

This is an interactive-style task with a non-adaptive grader. You must submit a file that implements:

```
int findNumber(int N);
```

Your function receives N and must return the hidden number H .

To ask about a number, call:

```
int ask(int x);
```

The value x must satisfy $1 \leq x \leq N$.

The function `ask(x)` returns:

- -1 if $H < x$;
- 0 if $H = x$;
- 1 if $H > x$.

You must include:

```
#include "grader.h"
```

Do not read from standard input or write to standard output.

Example Interaction

Suppose $N = 10$ and the hidden number is $H = 7$.

Your function could interact with the grader as follows:

Your function call	Grader returns	Reason
<code>ask(5)</code>	1	The hidden number is greater than 5.
<code>ask(8)</code>	-1	The hidden number is smaller than 8.
<code>ask(7)</code>	0	The hidden number was found.

After this, `findNumber(10)` should return 7.

Constraints

- $1 \leq N \leq 10^9$
- $1 \leq H \leq N$

You may not call `ask` more than 60 times.

Subtasks

Subtask 1 (30 points)

$$N \leq 50$$

Subtask 2 (70 points)

No additional constraints.

Scoring

For each test case, your solution receives full points if it returns the correct hidden number using at most 60 calls to `ask`.

Otherwise, it receives 0 points for that test case.

The final score is the minimum score over all test cases.

Sample Grader

For local testing, put your implementation in a file named `guess_number.cpp`, next to `grader.cpp` and `grader.h`, and compile:

```
g++ -std=c++20 -O2 -o guess_number guess_number.cpp grader.cpp
```

Then run:

```
./guess_number
```

The example grader reads the following input from standard input:

```
N H
```

For example:

```
10 7
```

If the answer returned by `findNumber` is incorrect, or if your function makes an invalid query, the grader prints:

```
0  
error message
```

If the answer is correct, the grader prints:

```
1  
Q
```

where Q is the number of calls to `ask`.