TRAINING COMPETITION OF THE
BULGARIAN EXTENDED NATIONAL TEAM
Bankya, 18 June 2025
group A

Problem AT1. HIDDEN

⏳0.1 sec. 💾 256 MB

There exists a number in the range $[1, n]$. You do not know it – it is hidden from you. To guess this number you can ask about its value. The jury will give you one of three answers:

- You have guessed the number.
- Your guess is greater than the hidden number.
- Your guess is lower that the hidden number.

However the jury is lazy – they are in no hurry to answer your queries. You will receive the answer of the i-th query after you ask the $(i + 1)$-th query. Despite the inconvenience their laziness imposes on you, you are determined to learn the value of this hidden number in at most t queries.

Implementation details

You must implement a function called:

```
void play(int n, int t);
```

that will attempt to figure out the hidden number. To ask a query you are supplied with the following function:

```
int guess(int x);
```

This function behaves in the following way: The first time you call the function, it returns -2 no matter what. Each next time you call the function, it returns either -1 if the previous guess was to greater than the hidden number, or $+1$ if the previous guess was less than the hidden number. If the previous guess was equal to the hidden number the communication will stop immeaditely and you will be awarded according to the *Subtasks and scoring* subsection.

Your program `hidden.cpp` must implement the function `play`. It may also contain other code and functions necessary for your implementation, but it must not contain the main function `main`. Additionally, you should not read from standard input or print to standard output. Your program must include the header file `hidden.h` using a preprocessor directive:

```
#include "hidden.h"
```

Subtasks and scoring

Subtask	**Points**	n	t	**Additional constraints**
0	0	–	–	Examples.
1	6	$= 5$	$= 6$	–
2	6	$= 7$	$= 5$	–
3	21	$= 50$	$= 9$	–
4	9	$= 350$	$= 13$	–
5	12	$= 2500$	$= 17$	–
6	20	$= 3 \times 10^5$	$= 27$	–
7	26	$= 10^9$	$= 100$	–

Points for a given subtask are only awarded if all tests provided for it are successfully passed with the sole exception of subtask 7. *Your score there is calculated according the table below.*

Condition	Points
$100 < Q$	0
$44 < Q \leq 100$	$\lfloor \frac{2250000}{Q^3} \rfloor$
$Q \leq 44$	26

Note that in some testcases the grader is **adaptive**.

Example

Contestant	Jury	Explanation
	`play(5, 6)`	Initiates a play. The hidden number is in the interval $[1, 5]$. You are allowed to use at most 6 queries.
`guess(3)`	`-2`	There is no previous query to answer. The return value is -2.
`guess(2)`	`+1`	Although the current guess is 2, this answer is for the 3 before it. It is less than the hidden number.
`guess(5)`	`+1`	2 is less than the hidden number, so the answer is $+1$.
`guess(4)`	`-1`	5 is greater than the hidden number, so the answer is -1.
`guess(1)`	$\emptyset$	4 is exactly the hidden number. Notice that you still have to make a query even after guessing the correct number. The communication will end by itself.

Local testing

You are provided with the file `Lgrader.cpp`, which you can compile together with your program in order to test it. When executed, the program will read the value of n, t from standard input and the value of the hidden number x. You will get the result of the communication and the number of used queries.