

# Zgadnij liczbę

Zadanie przykładowe

Short name: **gue**  
Time limit: **1 s**  
Memory limit: **64 MB**

Dana jest dwuosobowa gra „zgadnij liczbę”. Zasady są następujące: pierwszy gracz wybiera liczbę z przedziału  $[1, n]$  oraz podaje drugiemu graczowi  $n$  ( $1 \leq n \leq 10^6$ ). Drugi gracz musi zgadnąć wybraną liczbę. Do dyspozycji ma zapytania postaci „czy szukana liczba jest większa niż  $x$ ?” Napisz program, który pomoże drugiemu graczowi zgadnąć liczbę tak, by nie musiał użyć zbyt wielu zapytań.

## Komunikacja

Jest to zadanie interaktywne, to znaczy Twój program powinien używać biblioteki, która pozwala na zadawanie pytań pierwszemu graczowi. Aby użyć biblioteki, należy wpisać na początku programu:

- **C++:** `#include "guelib.h"`
- **Python:** `from guelib import init, isGreater, answer`

Biblioteka udostępnia następujące trzy funkcje:

- `init()`  
Funkcja ta powinna zostać wywoływana tylko raz, na początku działania programu. Rozpoczyna ona grę, a gracz pierwszy wymyśla szukaną liczbę. Zwraca liczbę  $n$ , podaną przez tego gracza.
  - **C++:** `int init();`
  - **Python:** `def init()`
- `isGreater(x)`  
Funkcja ta pyta pierwszego gracza, czy wybrana przez niego liczba jest większa od  $x$ . Zwraca `true` jeśli jest, `false` w przeciwnym wypadku. Podana liczba powinna należeć do przedziału  $[1, n]$ .
  - **C++:** `bool isGreater(int x);`
  - **Python:** `def isGreater(x)`
- `answer(x)`  
Jest to funkcja, którą należy wywołać dokładnie raz, na koniec działania Twojego programu. Służy do podania odpowiedzi pierwszemu graczowi, a liczba  $x$  jest zgadywaną przez nas wartością.
  - **C++:** `void answer(int x);`
  - **Python:** `def answer(x)`

Twój program **nie może** czytać żadnych danych (ani ze standardowego wejścia, ani z plików). **Nie może** również nic wypisywać do plików ani na standardowe wyjście. Może pisać na standardowe wyjście diagnostyczne (`stderr`) – pamiętaj jednak, że zużywa to cenny czas.

Po użyciu funkcji `answer` Twój program powinien natychmiastowo zakończyć działanie.

## Przykładowe wykonanie programu

| Funkcja                   | Wynik | Opis  |
|---------------------------|-------|---|
| <code>init()</code>       | 5     | Gra się rozpoczyna. Gracz pierwszy wybiera liczbę 4, a szukamy liczby w przedziale $[1, 5]$ , o czym jesteśmy poinformowani wartością funkcji <code>init()</code> . |
| <code>isGreater(4)</code> | false | Pytamy czy szukana liczba jest większa od 4. Nie jest ona większa (gdyż jest ona równa), więc otrzymujemy odpowiedź false.  |
| <code>isGreater(3)</code> | true  | Pytamy czy szukana liczba jest większa od 3. Jest ona większa, więc otrzymujemy odpowiedź true.   |
| <code>answer(4)</code>    | -     | Wiemy już, że szukaną liczbą jest 4, więc udzielamy odpowiedzi i kończymy program.  |

## Ocenianie

Żeby program dostał jakiegokolwiek punkty, musi on stosować się do zasad opisanych w sekcji komunikacji. Jeżeli naruszy on którąś z tych zasad, otrzyma werdykt „Błędna odpowiedź”.

Jeśli powyższe warunki zostaną spełnione, to program oceniany jest w następujący sposób.

Jeżeli zgadnięta liczba jest niepoprawna, nie otrzyma on punktów za dany test.

W przeciwnym wypadku niech  $k$  oznacza liczbę zapytań.

- Jeśli  $k$  nie przekracza 20, program dostanie pełną liczbę punktów.
- Jeśli  $k$  przekracza 30, ale nie 2000, program dostanie 50% punktów za test.
- Jeśli  $k$  przekracza 20, ale nie przekracza 30, to program otrzyma  $50 + (30 - k) \cdot 5$  punktów.
- Jeśli powyższe warunki nie są spełnione, program dostanie 0 punktów.

## Eksperymenty

W katalogu dławazaw znajdują się pliki, które pozwolą Ci przetestować poprawność formalną rozwiązania. Możesz tam znaleźć następujące pliki:

- **C++:** plik nagłówkowy `guelib.h`, bibliotekę `guelib.cpp` oraz przykładowy błędny program `gue.cpp`
- **Python:** bibliotekę `guelib.py` i przykładowy błędny program `gue.py`

Zwróć uwagę, że biblioteki te różnią się od tych, przy pomocy których będzie finalnie oceniane Twoje rozwiązanie, i służą jedynie do sprawdzenia poprawnej interakcji. Znajdując się w katalogu dławazaw, możesz standardowo skompilować i uruchomić `cyk.cpp` lub `cyk.py`. Przykładowa kompilacja w języku C++ to:

- `g++ -O3 -static gue.cpp guelib.cpp -std=c++11 -o gue`

Programy w Pythonie nie wymagają dodatkowych kroków do uruchomienia.

Tak otrzymany program wczytuje z wejścia liczbę  $n$ , a w kolejnym wierszu sekretną liczbę  $x$ .

Pamiętaj, że dostępna przykładowa biblioteka nie sprawdza, czy dane na wejściu są sformatowane poprawnie ani czy spełnione są ograniczenia z treści zadania.