

Guess the number

Example task

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.

Biblioteka

Jest to zadanie interaktywne, to znaczy Twój program będzie porozumiewał się z biblioteką. Aby użyć biblioteki, należy załączyć nagłówek `#include "gue.h"`. Biblioteka udostępnia następujące funkcje:

- `int 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.
- `bool isGreater(int x)` – funkcja ta pyta pierwszego gracza, czy wybrana przez niego liczba jest większa od x . Zwraca `true` jeśli jest, `false` w przeciwnym wypadku.
- `void answer(int x)` – funkcja ta zgaduje, jaka jest liczba pierwszego gracza. To kończy grę. Powinna być ona wywołana tylko raz, a po jej wykonaniu program powinien się zakończyć.

Kompilacja na swoim komputerze

Pliki z archiwum `gue_dla_zaw.zip` dostępnego w zakładce "Pliki" należy wypakować do folderu z kodem źródłowym programu. Aby program się skompilował należy załączyć nagłówek `#include "gue.h"`.

Program należy skompilować razem z biblioteką `gue_lib.cc`. Można to zrobić za pomocą polecenia:

```
g++ twoj_program.cpp gue_lib.cc -o twoj_program.
```

Wyjście

Twój program nie powinien pisać na standardowe wyjście (`stdout`) ani czytać ze standardowego wejścia (`stdin`). Dozwolone jest pisanie na standardowe wyjście diagnostyczne (`stderr`), lecz pamiętaj, że zabiera to cenny czas.

Przykład

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>	<code>false</code>	Pytamy czy szukana liczba jest większa od 4. Nie jest ona większa (gdyż jest ona równa), więc otrzymujemy odpowiedź <code>false</code> .
<code>isGreater(3)</code>	<code>true</code>	Pytamy czy szukana liczba jest większa od 3. Jest ona większa, więc otrzymujemy odpowiedź <code>true</code> .
<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 działać zgodnie z wymaganiami, czyli jako pierwszą wywołać funkcję `init`, jako ostatnią `answer` i każdą z nich dokładnie raz. Musi on również stosować się do zasad opisanych w sekcji wyjście.

Jeżeli naruszy on którąś z tych zasad, otrzyma werdykt WA.

Jeżeli 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.

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