

Zadanie: ZGA

Zgaduj zgadula

polish

ONTAK 2026, dzień 4. Dostępna pamięć: 256 MB. Limit czasu: 1.0 s.

01.07.2026

Antek i Jacek uwielbiają grać w grę *n-zgaduj zgadula*. Na początku gry Antek wybiera sobie pewną dodatnią liczbę całkowitą v mniejszą bądź równą n , a celem Jacka jest zgadnięcie tej liczby. Aby to osiągnąć Jacek może spytać się dowolnie wiele razy Antka czy liczba v należy do wybranego przez Jacka zbioru S . Po tych zapytaniach powinien zgadnąć liczbę którą wybrał Antek.

Niestety Antek ma brzydki zwyczaj kłamania, a Jacek wiedząc o tym wynegocjował sobie, że będzie miał dwie próby zgadnięcia wartości v oraz Antek nigdy nie skłamie odpowiadając na te dwa pytania. Antek ma jednak resztki honoru i nigdy nie skłamie dwa razy z rzędu na zadane przez Jacka pytanie o zbiór S .

Dzisiaj rano Antek z ubolewaniem odkrył, że Jacek się rozchorował i nie będzie w stanie z nim zagrać. W związku z tym chce zagrać z Tobą na tych samych zasadach.

Komunikacja

To zadanie jest interaktywne. Należy napisać program, który będzie komunikował się z Antkiem, używając do tego dostarczonej biblioteki. **Nie wolno wczytywać danych ani wypisywać danych korzystając z standardowych strumieni wejścia i wyjścia** – można jednak wypisywać dane na standardowe wejście diagnostyczne `cerr` – należy jednak pamiętać, że zużywa ono cenny czas. Aby użyć biblioteki, należy na początku swojego programu dopisać `#include "zgalib.h"`.

Biblioteka udostępnia trzy funkcje:

- `int dajN()` — Jako że nigdy wcześniej nie grałeś z Antkiem nie wiesz w którą wersję *zgaduj zgadula* będziecie grali (twoją ulubioną jest *1-zgaduj zgadula*). Funkcja ta powinna być wywołana dokładnie raz na samym początku twojego programu. Zwraca ona liczbę całkowitą n ($1 \leq n \leq 100\,000$) oznaczającą wersję *n-zgaduj zgadula* w którą będziecie grać.
- `int spytaj(const std::vector<int> &S)` — Funkcja ta służy do pytania Antka o zbiór S . Jeżeli zwróci ona 1 to znaczy, że Antek stwierdził, że v należy do zbioru S . W przeciwnym razie zwróci ona 0. **Uwaga:** jeżeli S zawiera tę samą liczbę kilka razy to program zakończy się statusem RTE.
- `int zgadnij(int x)` — Funkcja ta służy do zgadywania wartości v . Zwróci ona 1 jeżeli $x = v$ i 0 w przeciwnym wypadku.

Ponadto funkcje `spytaj` i `zgadnij` będą zwracać -1 jeżeli w trakcie trwania programu wydarzyła się jedna z poniższych rzeczy:

- Funkcja `spytaj` została wywołana po raz 90.
- Funkcja `zgadnij` została wywołana drugi raz (to wywołanie zwróci jeszcze poprawny wynik, ale wszystko po nim będzie zwracało -1).
- Funkcja `zgadnij` zwróciła 1 (tj. użytkownik zgadł poprawną liczbę).

Przykładowy przebieg programu

Antek wybrał $v = 3$.

Wywołana metoda	Wynik	Opis
<code>dajN()</code>	3	Antek chce grać w <i>3-zgaduj zgadula</i>
<code>spytaj({1, 2})</code>	1	Antek kłamie, że jego liczba znajduje się w tym zbiorze
<code>spytaj({3})</code>	1	Antek nie może teraz skłamać
<code>zgadnij(3)</code>	1	Jest to poprawna odpowiedź

Uwagi

1. Działanie niezgodne ze specyfikacją biblioteczki mogą skutkować działaniem losowym (tj. program może zwrócić dowolny z błędów w systemie `sio2`).
2. Biblioteczka nie musi ustalić wartości v na początku działania. Zagwarantowane jednak jest, że dla odpowiedzi które zostały zwrócone przez funkcje `spytaj` istnieje taki wybór v , że gdyby Antek wybrał je na początku to odpowiedzi na te pytania byłyby zgodne z opisaną wcześniej grą.

3. Próby wyjścia poza abstrakcje zadania mogą zakończyć się wyzerowaniem punktów za zadanie bądź nawet cały kontest (tj. nie należy nadmiernie kombinować z biblioteką).
4. W plikach do pobrania znajduje się przykładowa implementacja funkcji z biblioteki. Można ją uruchomić komendą

```
g++ -o zga zga.cpp zgalib.h -std=c++20
```

przy czym należy upewnić się, że plik `zga.cpp` i `zgalib.h` znajdują się w tym samym folderze.

Ocenianie

W zadaniu są 4 podzadania.

Podzadanie	Ograniczenia	Punkty
1	Wartość v jest ustalana przed rozpoczęciem programu. Antek mówi zawsze prawdę.	11
2	Wartość v jest ustalana przed rozpoczęciem programu. Jeżeli Antek może skłamać to z prawdopodobieństwem $1/2$ kłamie.	15
3	$n \leq 10000$	12
4	brak dodatkowych ograniczeń	62

Punktacja

Niech x będzie największą liczbą zapytań jakie wywoła program w danym podzadaniu, a t liczbą punktów za zadanie. Definiujemy wartość $s = \frac{90-x}{90-54}$.

- Jeżeli $x \leq 53$ to za dane podzadanie program otrzyma t punktów.
- Jeżeli $x \geq 90$ to za dane podzadanie program otrzyma 0 punktów.
- Jeżeli $53 < x < 90$ to za dane podzadanie program otrzyma

$$\left\lfloor \frac{\lfloor s^2 \cdot 80 \rfloor}{100} \cdot t \right\rfloor$$

punktów.

To znaczy, że liczba punktów maleje jak na poniższym wykresie:

