

Zadanie: ZAC

Punkty Giovanniego



Warsztaty ILO, Bonusy, Dostępna pamięć: 128 MB. Limit czasu: Limit 1 s.

19.06.2024

Giovanni jest zawodnikiem pewnych warsztatów czerwcowych. Zgłosił już rozwiązania do wszystkich zadań, ale wyniki testów do niektórych zadań pozostaną ukryte do końca konkursu.

Bardzo chciałby się dowiedzieć ile punktów ma za każde z zadań, żeby móc zdebugować te, na których nie ma wymarzonego maksa przed końcem konkursu.

Zaczął więc zadawać pytania dla Jury postaci "Czy mam więcej niż X punktów za zadanie XYZ?". Jury na początku na nie odpowiadało odpowiedzią "Tak" lub "Nie", ale zaczęło je już to irytować. Zaczęło więc złośliwie odpowiadać "Nie powiemy, ale masz więcej niż X na Y zadaniach".

Giovanni ma mało czasu do końca contestu, i będzie zadawał pytania Jury, aż pozna dokładną sumę punktów za tę rundę konkursu. Twoim zadaniem jest zadawać pytania dla Jury w imieniu Giovanniego.

Komunikacja

Jest to zadanie interaktywne, to znaczy Twój program powinien używać biblioteki, która pozwala na zadawanie pytań Jury. Aby użyć biblioteki, należy wpisać na początku programu: `#include "zaclib.h"`. Biblioteka udostępnia następujące cztery funkcje:

- `void init()`
Funkcja ta powinna zostać wywoływana tylko raz, na początku działania programu.
- `int getN()`
Zwraca liczbę całkowitą N ($1 \leq N \leq 10^6$) będącą liczbą zadań na koncieście, w którym bierze udział Giovanni. Zadania są ponumerowane liczbami całkowitymi od 1 do N .
- `int getMax()`
Zwraca liczbę całkowitą M ($1 \leq M \leq 10^9$), będąca maksymalną możliwą liczbą punktów za rozwiązanie zadania. Liczba punktów za każde zadanie jest liczbą całkowitą z zakresu od 1 do M .
- `int ileMamPunktow(int x)`
Zwraca liczbę zadań na których Giovanni ma więcej niż x punktów.
- `void odpowiedz(long long sumaPunktow)`
Funkcja ta powinna zostać wywołana dokładnie raz, jej jedynym argumentem powinna być suma punktów jaką Giovanni uzyskał na koncieście. Po jej wykonaniu twój program powinien natychmiast zakończyć wykonanie.

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.

Przykładowe wykonanie programu

Funkcja	Wynik	Opis
<code>init()</code>	-	Giovanni zgłasza wszystkie zadania. Sprawdzarka oceniła je i ustaliła, że dostał za nie kolejno 100 i 0 punktów.
<code>getN()</code>	2	Dowiadujesz się, że na koncieście są dokładnie 2 zadania.
<code>ileMamPunktow(0)</code>	1	Giovanni pyta, czy ma więcej 0 niż X punktów za ukryte zadanie. Jury odpowiada, że ma na dokładnie 1 zadaniu ma więcej niż 0 punktów.
<code>odpowiedz(100)</code>	-	Giovanni jest już pewny, że wyzerował drugie zadanie, więc odpowiada, że ma 100 punktów. Program kończy działanie.

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”.

Podzadanie	Ograniczenia	Punkty
1	$1 \leq M \leq 10^6$	25
2	$N = 1$	25
3	Brak dodatkowych ograniczeń	50

Eksperymenty

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

- plik nagłówkowy `zaclib.h`,
- bibliotekę `zaclib.cpp`,
- przykładowy błędny program `zac.cpp`

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 `zacdlazaw`, możesz skompilować rozwiązanie poleceniem: `g++ -O3 -static zac.cpp zaclib.cpp -std=c++20 -o zac`