Zadanie: SKA
Skarb

XXXI OI, etap III, dzień pierwszy. Plik źródłowy `ska.*` Dostępna pamięć: 256 MB. *10.04.2024*

Bitek znalazł skarb – skrzynię z pewną dodatnią liczbą monet x. Bajtosia, biorąc pod uwagę wymiary skrzyni, sprytnie wywnioskowała, że liczba monet w środku na pewno nie przekracza n. Chciałaby jednak dokładnie poznać, ile ich jest. Bitek nie chce jej tego powiedzieć, ale zgodził się zagrać z nią w pewną grę. Bajtosia może podać mu dodatnią liczbę całkowitą m, a on odpowie jej na jedno z dwóch pytań:

- Ile wynosi część całkowita z dzielenia przez m liczby monet w skrzyni?
- Ile wynosi reszta z dzielenia przez m liczby monet w skrzyni?

Zwróć uwagę, że to **Bitek wybiera**, na które pytanie odpowie, i komunikuje Bajtosi zarówno wybrane pytanie, jak i odpowiedź na to pytanie.

Bitek jest zbyt zajęty rozmyślaniem jak zainwestować pieniądze, więc zgodził się odpowiedzieć tylko na kilka pytań. Czy pomożesz Bajtosi mimo to umiejętnie zadać pytania i ustalić liczbę monet?

Interakcja

To zadanie jest interaktywne. Napisz program, który będzie zadawał Bitkowi pytania w imieniu Bajtosi i odgadnie x, używając do tego dostarczonej biblioteki. Aby użyć biblioteki, należy wpisać w swoim programie:

- C++: `#include "skalib.h"`
- Python: `from skalib import daj_ograniczenie, zapytaj, odpowiedz`

Twój program zostanie skompilowany/uruchomiony za pomocą komendy:

- C++: `g++ -O3 -static ska.cpp skalib.cpp -std=c++20`
- Python: `python3 ska.py`

Aby poznać ustalone przez Bajtosię ograniczenie górne na liczbę monet w skrzyni, można wywołać funkcję:

- C++: `unsigned long long daj_ograniczenie()`
- Python: `daj_ograniczenie() -> int`

Aby zadawać Bitkowi pytania, należy wywoływać funkcję:

- C++: `std::pair<char, unsigned long long> zapytaj(unsigned long long m)`
- Python: `zapytaj(m : int) -> tuple[str, int]`

Każde wywołanie tej funkcji daje jedną z dwóch możliwych odpowiedzi:

- parę $('/', w)$, oznaczającą, że $\lfloor x/m \rfloor = w$ (*część całkowita z dzielenia x przez m jest równa w*),
- parę $('\%', w)$, oznaczającą, że $x \equiv w \pmod{m}$ (*reszta z dzielenia x przez m jest równa w*).

Żeby powiedzieć Bajtosi, ile monet znajduje się w skrzyni, należy wywołać funkcję:

- C++: `void odpowiedz(unsigned long long x)`
- Python: `odpowiedz(x : int) -> ()`

z ustaloną wartością parametru x. Po wywołaniu tej funkcji należy od razu zakończyć program.

Twój program nie może otwierać żadnych plików ani używać standardowego wejścia i wyjścia. Może on korzystać ze standardowego wyjścia diagnostycznego (`stderr`), jednak pamiętaj, że zużywa to cenny czas.

Przykładowy przebieg programu

Test 0 w SIO odpowiada przypadkowi, w którym $n = 100$ i $x = 48$.

Wywołana funkcja	Wynik	Opis
`daj_ograniczenie()`	100	$n = 100$
`zapytaj(9)`	$('\%', 3)$	$x \equiv 3 \pmod{9}$
`zapytaj(10)`	$('/', 4)$	$\lfloor x/10 \rfloor = 4$
`odpowiedz(48)`	—	$x = 48$ to jedyna wartość zgodna z powyższymi odpowiedziami

Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025”.

Ocenianie

Aby zdobyć maksymalną punktację, funkcję `zapytaj` wolno wywołać **co najwyżej cztery razy**. Jednak możliwe jest uzyskanie częściowej punktacji również w przypadku przekroczenia tego limitu (patrz tabela poniżej).

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Ograniczenia	Punkty
1	$0 \leq x < 2^4$	5
2	$0 \leq x < 2^8$	6
3	$0 \leq x < 2^{16}$	9
4	$0 \leq x < 2^{32}$	30
5	$0 \leq x < 2^{64}$	50

Niech q oznacza liczbę wywołań funkcji `zapytaj`. Punkty za test przyznawane są wtedy zgodnie z poniższą tabelą.

$0 \leq q \leq 4$	100% maksymalnej punktacji za test
$q = 5$	90%
$q = 6$	80%
$q = 7$	60%
$q = 8$	50%
$q = 9$	40%
$10 \leq q \leq 18$	$(37 - 3 \cdot (q - 10))\%$
$19 \leq q \leq 30$	10%
$q > 30$	0%

Jeżeli funkcja `zapytaj` zostanie wywołana z parametrem równym 0 lub parametr x funkcji `odpowiedz` będzie inny niż oczekiwany, Twój program otrzyma werdykt *Błędna odpowiedź* i 0 punktów za test.

W niektórych testach biblioteka może być adaptacyjna – tzn. nie ustalać na początku wartości x i nie ustalać na początku, jak zareaguje na każde możliwe pytanie. Możesz założyć, że w dowolnym momencie interakcji istnieje co najmniej jedna liczba x zgodna z dotychczas udzielonymi odpowiedziami.

Eksperymenty

W sekcji *Pliki i testy* w SIO możesz znaleźć przykładową implementację biblioteki. Ta implementacja daje co drugą odpowiedź '`%`', a co drugą '`/`'. Biblioteka używana do sprawdzania rozwiązań może wybierać inne odpowiedzi na pytania.

Uwaga: W tym zadaniu nie są dostępne uruchomienia próbne w SIO ani skrypt `ocen` na komputerach.

Wskazówka: W języku C++, w regulaminowym kompilatorze jest dostępny 128-bitowy typ całkowity ze znakiem o nazwie `__int128`. Zwracamy uwagę, że wartości tego typu nie można w standardowy sposób wczytywać ani wypisywać.