

# Zadanie: ZER

## Zera i jedyнки



XXIX OI, etap III, dzień próbny. Plik źródłowy zer.\* Dostępna pamięć: 256 MB.

5.04.2022

Jest sobie pewien ciąg  $a_0, \dots, a_{n-1}$  złożony z liczb 0 i 1. Nie znasz tego ciągu, możesz jednak zapytać o sumę dowolnej pary różnych elementów ciągu. Twoim zadaniem jest odgadnąć ciąg za pomocą małej liczby takich pytań.

## Komunikacja

Zaimplementuj program, który odgadnie ciąg zer i jedynek, korzystając z dostarczonej biblioteki odpowiadającej na podane pytania. Aby użyć biblioteki, należy wpisać w swoim programie:

- C/C++: `#include "zerlib.h"`
- Python: `from zerlib import daj_n, suma, odpowiedz`

Biblioteka udostępnia następujące funkcje:

- `daj_n()` – Funkcja zwraca liczbę całkowitą  $n$ , oznaczającą długość nieznanego ciągu zer i jedynek.
- `suma(i, j)` – Funkcja daje w wyniku  $a_i + a_j$ , jeśli  $0 \leq i, j < n$  oraz  $i \neq j$ . W przeciwnym razie jej wywołanie powoduje błędną odpowiedź.
- `odpowiedz(a)` – Ta funkcja pozwala nam zgłosić wynikowy ciąg. Należy ją wykonać dokładnie raz.

Przyjmuje ona tablicę  $a[0..n-1]$  (daną jako wektor `std::vector<int>` w języku C++ lub lista w języku Python), reprezentującą ciąg wynikowy  $a[0], \dots, a[n-1]$ . Po wykonaniu tej funkcji program zostanie automatycznie zakończony. Jeśli podany w funkcji ciąg będzie błędny lub jego długość będzie inna niż  $n$ , spowoduje to błędną odpowiedź.

Biblioteka **nie musi** generować ciągu  $a_0, \dots, a_{n-1}$  na początku interakcji z Twoim programem. Może ona w trakcie interakcji zmieniać elementy ciągu, o ile nowe elementy są nadal spójne z wynikami zwróconymi przez dotychczasowe wywołania funkcji `suma`.

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.

Rozwiązanie będzie kompilowane wraz z biblioteką następującymi poleceniami:

- C++: `g++ -O3 -static -std=c++17 zerlib.cpp zer.cpp`
- Python: `python3 zer.py`

**Uwaga:** Podane na górze ograniczenie pamięci dotyczy tylko Twojego rozwiązania, a zatem nie wlicza pamięci wykorzystywanej przez bibliotekę.

## Ocenianie

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

Podzadanie	Warunki	Liczba punktów
1	$3 \leq n \leq 1000$	50
2	$3 \leq n \leq 200\,000$	50

Jeśli  $m$  to maksymalna liczba wywołań funkcji `suma`, które Twój program wykonał w jednym przypadku testowym, to Twoje rozwiązanie otrzyma następujący procent punktów za test (odpowiednio przeskalowany w przypadku przekroczenia połowy limitu czasowego):

Liczba wywołań	Procent punktów
$m \leq n$	100% punktów za test
$m = n + 1$	80% punktów za test
$m \leq n^2 - n$	50% punktów za test
$m > n^2 - n$	0% punktów za test (werdykt <i>Błędna odpowiedź</i> )

## Przykładowy przebieg programu

Poniżej przedstawiono przykładowy przebieg programu dla testu przykładowego.

Wywołanie	Wynik	Opis
daj_n	5	$n = 5$
suma(0, 1)	1	$a_0 + a_1 = 1$
suma(1, 2)	1	$a_1 + a_2 = 1$
suma(3, 4)	2	$a_3 + a_4 = 2$ , a więc $a_3 = a_4 = 1$
suma(0, 3)	2	$a_0 + a_3 = 2$ , a więc $a_0 = 1$ , skąd wynika też, że $a_1 = 0$ i $a_2 = 1$
odpowiedz({1,0,1,1,1})	–	Prawidłowa odpowiedź z użyciem $m = 4 \leq n = 5$ pytań, 100% punktów za test

Testy „ocen”:

1ocen:  $n = 1000$ ,  $a = 0^{500}1^{500}$ ;

2ocen:  $n = 200\,000$ ,  $a = (01)^{100\,000}$ .

## Eksperymenty

Przykładowe **błędne** rozwiązania wraz z przykładowymi bibliotekami znajdują się w folderze **d1azaw**. Biblioteki mogą różnić się zachowaniem od tych używanych do ostatecznej oceny rozwiązań i nie spełniać założeń zadania. Mają one jedynie pokazać sposób interakcji z programem.

Twoje rozwiązanie skompilowane z przykładową biblioteką wczytuje ze standardowego wejścia opis ciągu – liczbę  $n$ , a następnie kolejne liczby  $a_0, \dots, a_{n-1}$ , oddzielone odstępami – następnie używa ich do odpowiadania na pytania `suma` z Twojego programu, a na końcu wypisuje na standardowe wyjście ciąg przekazany w wywołaniu funkcji `odpowiedz`.