

# Zadanie: TEA

## Teatr [B]



POTYCZKI ALGORYTMICZNE

Potyczki Algoritmiczne 2018, runda czwarta. Limity: 256 MB, 6 s.

13.12.2018 - 14.12.2018

**Uwaga!** To jest zadanie rozproszone. Zanim zaczniesz je rozwiązywać, zapoznaj się z informacjami dotyczącymi tego typu zadań dostępnymi w serwisie internetowym zawodów.

Najnowszy budynek Teatru Awangardowego w Bajtocji od samego początku wzbudza kontrowersje. Architekt zaprojektował w nim widownię, która ma  $N$  rzędów, ale w każdym z nich tylko jedno miejsce. Ma to na celu przede wszystkim zwracać uwagę widzów na problem wyobcowania jednostki we współczesnym zglobalizowanym świecie, choć niezyczliwi twierdzą, że to kwestia działki budowlanej, która okazała się mieć nietypowe wymiary.

Bajtoccy melomani są z natury drażliwi – jeśli ktoś będzie miał pecha usiąść za osobą wyższą od siebie, na pewno skończy się to kłótnią. Co więcej, dla każdej pary rzędów  $(i, j)$  takich że  $i < j$ , a osoba w rzędzie  $i$  ma wzrost większy niż osoba w  $j$ , można spodziewać się widowiskowej awantury w czasie lub po zakończeniu przedstawienia.

Mając dany wzrost wszystkich osób siedzących danego wieczoru na sali, znajdź liczbę kłótni.

Zwróć uwagę na nietypowy limit na łączny rozmiar wiadomości wysłanych z pojedynczej instancji.

## Komunikacja z biblioteką

Aby móc rozwiązywać zadanie, należy na początku programu dopisać dyrektywę

```
#include "teatr.h"
```

Dodana biblioteka udostępnia następujące funkcje:

- `int GetN()`; – zwraca liczbę rzędów na sali.
- `int GetElement(int k)`; – dla podanego  $0 \leq k < \text{GetN}()$  zwraca liczbę całkowitą, oznaczającą wzrost osoby siedzącej w  $k$ -tym rzędzie.

## Wyjście

Dokładnie jedna instancja powinna wypisać jedną liczbę całkowitą – liczbę awantur, do jakich może dojść tego wieczoru na sali.

## Ograniczenia

- $1 \leq \text{GetN}() \leq 100\,000\,000$
- $1 \leq \text{GetElement}(k) \leq 1\,000\,000$
- Każdy test będzie uruchomiony na 100 instancjach komputerów.
- Podany limit pamięci dotyczy pojedynczej instancji.
- Limit liczby wiadomości wychodzących z pojedynczej instancji: 1000.
- **Limit łącznego rozmiaru wiadomości wychodzących z pojedynczej instancji: 8 KB.**
- Wywołanie funkcji bibliotecznych zajmuje średnio nie dłużej niż 0.035 mikrosekundy.

## Podzadania

W niektórych grupach testów spełniony jest dodatkowy warunek `GetElement(k) ≤ 5`.

## Testy przykładowe

Po wysłaniu rozwiązania od razu poznasz wyniki sprawdzania na poniższym teście przykładowym, uruchomionym na 100 instancjach.

Dla poniższego przebiegu programu:

Wywołanie funkcji	Zwrócona wartość
GetN()	5
GetElement(0)	5
GetElement(1)	2
GetElement(2)	4
GetElement(3)	4
GetElement(4)	3

poprawnym wynikiem jest:

6

**Wyjaśnienie do przykładu:** Kłótnia może wybuchnąć między parami (0, 1), (0, 2), (0, 3), (0, 4), (2, 4) oraz (3, 4).

## Eksperymenty

W dziale *Pliki* możesz znaleźć przykładową bibliotekę testującą. Składa się ona z plików `teatr.h` oraz `teatr.cpp`. Znajduje się tam też przykładowy (błędny) program `tea_wrong.cpp` korzystający z interfejsu udostępnianego przez bibliotekę.

Aby skompilować Twój program `tea.cpp` z biblioteką, należy wydać następującą komendę (zakładamy, że `rpa` jest zdefiniowane jak w poradniku do zadań rozproszonych):

```
rpa build --source tea.cpp --library teatr.cpp
```

Tak skompilowany program wczytuje test z przykładowego wejścia. Pierwszy wiersz powinien zawierać pojedynczą liczbę całkowitą  $N$  – liczbę rzędów na sali. Drugi i ostatni wiersz powinien zawierać  $N$  liczb całkowitych  $a_i$ , gdzie  $a_i$  dla  $i = 0, 1, \dots, N - 1$  oznacza wzrost osoby siedzącej w  $i$ -tym rzędzie.

Dla ułatwienia, test przykładowy został przetłumaczony do powyższego formatu (plik `tea0a.in` w archiwum powyżej).