



# Zadanie: KAN

## Kanapka

Potyczki Algoritmiczne 2018, runda próbna. Limity: 128 MB, 2-3 s.

06.12.2018 - 07.12.2018

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

Niniejsze zadanie pojawiło się także w rundzie próbnej konkursu *Distributed Google Code Jam 2015*.

Żona przygotowała Bajtazarowi długą kanapkę. Kanapka składa się z  $N$  części różniących się pod względem użytych składników. Bajtazar wie dokładnie, które części będą mu smakować i jak bardzo: dla każdej części określił jej smakowitość liczbą całkowitą. Im większa jest ta liczba, tym bardziej Bajtazar chciałby zjeść tę część; ujemna smakowitość oznacza, że Bajtazar wolałby nie jeść danej części kanapki.

Oczywiście Bajtazar najchętniej zjadłby jedynie części o dodatniej smakowitości. Wyłamywanie środkowych części kanapki jest jednak niegrzeczne i z pewnością nie spodobałoby się żonie. Bajtazar postanowił więc, że zje po trosze z początku i z końca kanapki tak, aby sumaryczna smakowitość zjedzonych części była jak największa. W szczególności, Bajtazar dopuszcza zjedzenie całej kanapki bądź niezjedzenie ani kawałka.

## Dane wejściowe

Twój program nie może używać standardowego wejścia. Zamiast tego dostęp do tablicy będzie odbywać się za pośrednictwem dostarczonej biblioteki interaktywnej. Aby jej użyć, wpisz w swoim programie:

```
#include "kanapka.h"
```

Biblioteka udostępni dwie funkcje:

- `GetN()` – zwraca  $N$  ( $1 \leq N \leq 5 \cdot 10^8$ ) – liczbę części kanapki.
- `GetTaste(i)` – zwraca liczbę całkowitą z przedziału  $[-10^9, 10^9]$  oznaczającą smakowitość  $i$ -tej części kanapki. Części kanapki są ponumerowane liczbami od 0 do  $N - 1$ .

Funkcjom tym odpowiadają następujące deklaracje:

```
long long GetN();  
long long GetTaste(long long i);
```

W dziale *Pliki* w systemie SIO2 znajduje się archiwum zawierające przykładowe pliki bibliotek oraz (niepoprawne) rozwiązania ilustrujące sposób ich użycia.

## Wyjście

Twój program powinien wypisać na wyjście jedną liczbę – maksymalną sumaryczną smakowitość pewnego początkowego i końcowego fragmentu kanapki.

## Komunikacja

Podczas oceny Twojego programu system sprawdzający uruchomi jednocześnie wiele jego instancji, każdą na osobnym komputerze. Instancje powinny komunikować się za pomocą biblioteki `message`. W tym celu w programie w języku C bądź C++ należy umieścić wiersz:

```
#include "message.h"
```

Instrukcja dotycząca używania tej biblioteki jest dostępna w serwisie internetowym zawodów w zakładce *Zadania rozproszone*.

## Ograniczenia liczby i rozmiaru wysyłanych wiadomości

- Liczba wiadomości wysłanych przez pojedynczą instancję nie może przekroczyć 1000.
- Sumaryczny rozmiar wiadomości wysłanych przez jedną instancję nie może przekroczyć 8MB.

## Przykładowy przebieg programu

Dla przebiegu programu:

Wywołanie funkcji	Zwrócona wartość
GetN();	7
GetTaste(0);	10
GetTaste(1);	-2
GetTaste(2);	5
GetTaste(3);	-4
GetTaste(4);	3
GetTaste(5);	-5
GetTaste(6);	1

poprawnym wynikiem jest:

14

## Testy przykładowe

Każdy z tych testów będzie uruchomiony na 10 komputerach.

- 0a: test przykładowy z treści zadania.
- 0b:  $N = 2n + 2$ , gdzie  $n = 1\,000\,000$ . Smakowitości kolejnych części tworzą następujący ciąg:

$0, -1, 2, -3, \dots, (n-2), -(n-1), n, -n, (n-1), -(n-2), \dots, 3, -2, 1, 0.$