

# **Zadania treningowe do I etapu Olimpiady o Diamentowy Indeks AGH**

Przedstawione zadania są zadaniami treningowymi i **nie będą oceniane**. Właściwe zadania konkursowe będą dostępne po zalogowaniu do systemu OIOIOI w dniach 23-25.10.2019.

# Zadanie testowe 1

Dana jest liczba całkowita  $n$ . Proszę napisać program, który wylicza ostatnią **niezerową** cyfrę liczby  $n!$ .

## Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajduje się jedna liczba całkowita  $n$  ( $0 \leq n < 10^{10}$ ).

## Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać jedną cyfrę będącą rozwiązaniem.

## Przykład

Dla danych wejściowych:

10

poprawną odpowiedzią jest:

8

## Zadanie testowe 2

Dane są dwie liczby naturalne,  $m$  i  $n$ . Proszę napisać program, który wyznacza sumę  $n$  kolejnych cyfr *po przecinku* rozwinięcia dziesiętnego liczby  $\sqrt{m}$ .

### Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajdują się dwie liczby naturalne,  $m$  i  $n$  ( $1 \leq m \leq 10^8$ ,  $1 \leq n < 100$ ).

### Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać jedną liczbę będącą rozwiązaniem.

### Przykład

Dla danych wejściowych:

2 4

poprawną odpowiedzią jest:

11

## Zadanie testowe 3

Proszę napisać program, który **dekompresuje** string. Kompresja polega na zastąpieniu powtarzających się sekwencji liter przez napis `[DX]`, gdzie `D` jest licznikiem powtórzeń (pojedynczą cyfrą) niepustej sekwencji `X`. `X` może również zawierać skompresowane podciągi. Na przykład `cababababababc` można zapisać jako `c[6ab]c` albo `c[2[3ab]]c`.

### Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajduje się skompresowany niepusty string o długości co najwyżej 50 znaków i zawierający: małe litery alfabetu łacińskiego ('a' do 'z'), cyfry ('1' do '9') i nawiasy ('[', ']').

### Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać zdekompresowany string. String po dekompresji zawiera co najwyżej 1000 znaków będących małymi literami alfabetu łacińskiego ('a' do 'z').

### Przykład

Dla danych wejściowych:

```
c[6ab]c
```

poprawną odpowiedzią jest:

```
cababababababc
```

## Zadanie testowe 4

Pewnych liczb nie można przedstawić jako sumy elementów spójnego fragmentu ciągu Fibonacciego ( $f_0 = f_1 = 1; f_n = f_{n-1} + f_{n-2}$  dla  $N > 1$ ), na przykład 9, 14, 15, 17, 22. Proszę napisać program, który wczytuje liczbę naturalną  $N$  i wypisuje następną taką liczbę większą od  $N$ .

### Wejście

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba naturalna  $N$  ( $0 < N < 10^6$ ).

### Wyjście

Program powinien wypisać na standardowe wyjście jedną liczbę.

### Przykład

Dla danych wejściowych:

17

poprawną odpowiedzią jest:

22

## Zadanie testowe 5

Dana jest  $N$  elementowa tablica  $T$  zawierająca liczby naturalne większe od 1. Po tablicy możemy przemieszczać się według następującej zasady: z pola o indeksie  $i$  możemy przeskoczyć na pole o indeksie  $i + k$ , jeżeli  $k$  jest czynnikiem pierwszym liczby  $T[i]$  mniejszym od  $T[i]$ . Proszę napisać program, który wyznacza i wypisuje na standardowe wyjście minimalną liczbę skoków koniecznych aby przejść z pola o indeksie 0 na pole o indeksie  $N - 1$ . Jeżeli powyższe przejście nie jest możliwe program powinien wypisać -1.

### Wejście

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba całkowita  $N$  oznaczająca rozmiar tablicy  $T$ . W każdym z kolejnych  $N$  wierszy znajduje się jedna liczba naturalna większa od 1 - kolejny element tablicy  $T$ .

### Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać jedną liczbę całkowitą, oznaczającą minimalną liczbę skoków koniecznych aby przejść z pola o indeksie 0 na pole o indeksie  $N - 1$ . Jeżeli powyższe przejście nie jest możliwe program wypisuje -1.

### Ograniczenia

$1 < N \leq 1000000$ , dla każdego elementu tablicy,  $T[i]$ , zachodzi  $1 < T[i] \leq 100000$ .

### Przykład

Dla danych wejściowych:

9  
10  
8  
25  
6  
12  
21  
6  
4  
5

poprawną odpowiedzią jest:

2

## Zadanie testowe 6

Dany jest ciąg zer i jedynek rozpoczynający się od jedynki. Proszę napisać program, który odpowiada na pytanie na ile sposobów można pociąć ten ciąg na kawałki, z których każdy reprezentuje zapisaną dwójkowo liczbę pierwszą, a powstały ciąg liczb pierwszych jest rosnący. Każdy kawałek musi rozpoczynać się od jedynki, a jego długość nie może być większa od 32. Na przykład dla ciągu 11111101 możliwe są dwa podziały spełniające warunki zadania, czyli: 11–111101 (dziesiętnie 3 i 61) oraz 111–11101 (dziesiętnie 7 i 29).

### Wejście

W pierwszym wierszu standardowego wejścia znajduje się ciąg zer i jedynek (bez spacji) o długości mniejszej niż 255.

### Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać liczbę sposobów, na które można pociąć ciąg wejściowy.

### Przykład

Dla danych wejściowych:

11111101

poprawną odpowiedzią jest:

2