

# AtCoder Beginner Contest 101

## Problema C: *Minimization*

---

Prof. Edson Alves - UnB/FGA

## **AtCoder Beginner Contest 101 – Problem C: Minimization**

---

# Problema

There is a sequence of length  $N$ :  $A_1, A_2, \dots, A_N$ . Initially, this sequence is a permutation of  $1, 2, \dots, N$ .

On this sequence, Snuke can perform the following operation:

- Choose  $K$  consecutive elements in the sequence. Then, replace the value of each chosen element with the minimum value among the chosen elements.

Snuke would like to make all the elements in this sequence equal by repeating the operation above some number of times. Find the minimum number of operations required. It can be proved that, under the constraints of this problem, this objective is always achievable.

## Constraints

- $2 \leq K \leq N \leq 100000$
- $A_1, A_2, \dots, A_N$  is a permutation of  $1, 2, \dots, N$

## Input

Input is given from Standard Input in the following format:

$$\begin{array}{c} N \quad K \\ A_1 \quad A_2 \quad \dots \quad A_N \end{array}$$

## Output

Print the minimum number of operations required.

# Exemplo de entradas e saídas

## Exemplo de Entrada

4 3  
2 3 1 4

3 3  
1 2 3

8 3  
7 3 1 8 4 6 2 5

## Exemplo de Saída

2

1

4

- O primeiro fato a ser observado é que é possível modificar, a cada operação, no máximo  $K - 1$  elementos
- O segundo fato importante é que o elemento mínimo da sequência é sempre o número 1 (um)
- Deve ficar claro que o objetivo não é realizar as operações, mas sim contar o mínimo necessário para completar a transformação
- Do primeiro fato apresentado segue que são necessárias pelo menos

$$\left\lceil \frac{N - 1}{K - 1} \right\rceil$$

operações, pois é preciso modificar os  $N - 1$  elementos restantes,  $K - 1$  por vez

- Observe também que a sequência de intervalos  $I_0, I_1, \dots, I_{\lceil \frac{N-1}{K-1} \rceil}$  cobre toda a sequência, onde

$$I_i = i(K-1) + 1, i(K-1) + 2, \dots, i(K-1) + K$$

- Com esta divisão da sequência, localize o intervalo  $I_k$  que contém o elemento 1
- Então aplique  $k$  operações nos intervalos  $I_k, I_{k-1}, \dots, I_0$ , nesta ordem
- Agora aplique a operação nos intervalos  $I_{k+1}, I_{k+1}, \dots, I_{\lceil \frac{N-1}{K-1} \rceil}$
- Logo, é possível finalizar o processo com  $\lceil \frac{N-1}{K-1} \rceil$  operações

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     ios::sync_with_stdio(false);
8
9     int N, K;
10    cin >> N >> K;
11
12    auto ans = (N - 1 + K - 2) / (K - 1);
13
14    cout << ans << '\n';
15
16    return 0;
17 }
```