

## Caching Offline

Limite de tempo: 1s  
Limite de memória: 256MB

Autor:

Dado um sistema com memória principal dividida em  $k$  frames, deseja-se gerenciar uma sequência de requisições de páginas de modo a minimizar falhas de memória (*cache misses*).

Considere que a sequência de páginas referenciadas  $(v_0, v_1, \dots, v_{n-1})$  é conhecida a priori, onde cada  $v_i$  representa o identificador de uma página. Inicialmente, a cache (memória principal) encontra-se vazia.

O objetivo é projetar um algoritmo que determine a **menor quantidade possível de falhas de página** para a sequência dada. Uma falha ocorre sempre que uma página referenciada não está presente em um dos  $k$  frames disponíveis. Se a cache estiver cheia e uma nova página precisar ser carregada, uma das páginas residentes deve ser substituída.

### Entrada

A entrada é composta por duas linhas.

A primeira linha contém dois inteiros:  $k$  ( $1 \leq k \leq 1000$ ), representando a quantidade de frames disponíveis na memória, e  $n$  ( $1 \leq n \leq 1000$ ), representando o número total de referências na sequência de páginas.

A segunda linha contém  $n$  inteiros positivos  $v_i$  ( $1 \leq v_i \leq 1000$ ), representando os identificadores de cada página na sequência de referências.

### Saída

A saída deve conter uma única linha.

Deve ser impresso um único inteiro representando o **menor número de falhas de página** (*cache misses*) que podem ser obtidos para a sequência e capacidade de memória fornecidas.

### Exemplo

| Entrada             | Saída |
|---------------------|-------|
| 1 3                 | 3     |
| 1 2 3               |       |
| 2 5                 | 3     |
| 1 2 3 1 1           |       |
| 5 10                | 5     |
| 1 2 3 4 5 5 4 3 2 1 |       |

### Notas

No primeiro caso de teste, temos  $k = 1$  frame e uma sequência de 3 referências:  $(1, 2, 3)$ . Como a memória só pode comportar uma página por vez e todas as referências são para páginas diferentes, cada nova requisição resultará em uma falha (*miss*) para carregar a página atual e substituir a anterior. Portanto, o total de falhas é 3.

No segundo caso, a memória possui  $k = 2$  frames e a sequência é  $(1, 2, 3, 1, 1)$ .

- As páginas 1 e 2 são carregadas inicialmente, causando 2 falhas.

- Quando a página 3 é requisitada, a memória está cheia, com as páginas  $\{1, 2\}$ . Substitui-se a página 2 pela 3,  $\{1, 3\}$ .
- As referências subsequentes para a página 1 já estão na memória.
- O número mínimo de falhas é 3.

No terceiro caso, com  $k = 5$  frames e a sequência  $(1, 2, 3, 4, 5, 5, 4, 3, 2, 1)$ , o número de páginas distintas na sequência é exatamente 5. Como a capacidade da memória é suficiente para manter todas as páginas únicas da sequência simultaneamente, ocorrem apenas as 5 falhas iniciais para o preenchimento da cache. Uma vez que todas as páginas estão carregadas, as referências de volta  $(5, 4, 3, 2, 1)$  resultam todas em acertos (*hits*). O total de falhas é 5.