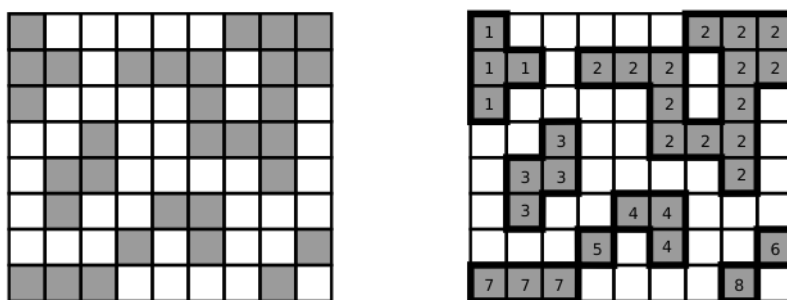


Manchas de pele

Nome do arquivo: `manchas.c`, `manchas.cpp`, `manchas.pas`, `manchas.java`, `manchas.js`, `manchas_py2.py` ou `manchas_py3.py`

O laboratório de dermatologia da Linearlândia está implementando um software para contar o número de manchas presentes numa imagem digital de N por M pixels. Cada pixel na imagem é preto ou branco e dois pixels pretos distintos A e B pertencem à mesma mancha se e somente se: existir uma sequência de pixels $[P_1, P_2, \dots, P_k]$, onde $k \geq 2$, $A = P_1$, $B = P_k$ e para todo $1 \leq i < k$, P_i é ortogonalmente adjacente a P_{i+1} (P_i imediatamente acima, abaixo, à esquerda ou à direita de P_{i+1}).



A figura acima, para $N = 8$ e $M = 9$, ilustra uma imagem digital onde existem oito manchas. Dada a imagem, seu programa deve contar o número de manchas presentes.

Entrada

A primeira linha da entrada contém dois inteiros N e M , representando, respectivamente, o número de linhas e colunas da imagem. As N linhas seguintes contêm, cada uma, M inteiros P representando os pixels da imagem.

Saída

Seu programa deve imprimir uma linha contendo um inteiro, o número de manchas na imagem.

Restrições

- $1 \leq N \leq 1000$
- $1 \leq M \leq 1000$
- O valor de P é 1, representando um pixel preto, ou 0, representando um pixel branco.

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 10 pontos, $N = M = 2$.
- Para um conjunto de casos de testes valendo outros 20 pontos, $N = 1$.
- Para um conjunto de casos de testes valendo outros 20 pontos, $N, M \leq 100$.
- Para um conjunto de casos de testes valendo outros 50 pontos, nenhuma restrição adicional (*Atenção, para essa parcial, não é recomendada uma implementação recursiva!*)

Exemplo de entrada 1 8 9 1 0 0 0 0 0 1 1 1 1 1 0 1 1 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 0	Exemplo de saída 1 8
Exemplo de entrada 2 1 1 0	Exemplo de saída 2 0
Exemplo de entrada 3 1 10 0 0 1 0 1 1 1 0 1 0	Exemplo de saída 3 3