

Complete a função `resolveMeFirst` para calcular a soma de dois números inteiros.

Função protótipo:

`int resolveMeFirst (int a, int b);`

Onde,

- `a` é a primeira entrada inteira.
- `b` é a segunda entrada inteira

Retornar valores

- soma dos dois inteiros acima

Entrada de amostra

```
a = 2
b = 3
```

Saída de amostra

```
5
```

Explicação

A soma dos dois inteiros *a* e *b* é calculado como: $2 + 3 = 5$

Dada uma matriz de números inteiros, encontre a soma de seus elementos.

Por exemplo, se a matriz $ar = [1, 2, 3]$, então retornar $1 + 2 + 3 = 6$

Descrição da função

Complete a função `simpleArraySum` no editor abaixo. Ele deve retornar a soma dos elementos da matriz como um número inteiro.

`simpleArraySum` possui o (s) seguinte (s) parâmetro (s):

- `ar` : uma matriz de números inteiros

Formato de entrada

A primeira linha contém um número inteiro n , denotando o tamanho da matriz.

A segunda linha contém n números inteiros separados por espaço que representam os elementos da matriz.

Restrições

$$0 < n, ar[i] \leq 1000$$

Formato de saída

Imprima a soma dos elementos da matriz como um único inteiro.

Entrada de amostra

```
6
1 2 3 4 10 11
```

Saída de amostra

```
31
```

Explicação

Nós imprimir a soma de elementos do array: $1 + 2 + 3 + 4 + 10 + 11 = 31$

Alice e Bob criaram um problema para o HackerRank. Um revisor classifica os dois desafios, atribuindo pontos em uma escala de 1 até três categorias: clareza do problema, originalidade e dificuldade. 100

Definimos a classificação para o desafio de Alice ser o triplo a , e a classificação para o desafio de Bob para ser o triplo b .

$$a = (a[0], a[1], a[2]) \quad b = (b[0], b[1], b[2])$$

Sua tarefa é encontrar seus pontos de comparação comparando a com b , com a e com b . $a[0]b[0]a[1]b[1]a[2]b[2]$

- Se $a[i] > b[i]$, então Alice recebe um ponto.
- Se $a[i] < b[i]$, então Bob recebe um ponto.
- Se $a[i] = b[i]$, então nenhuma pessoa recebe um ponto.

Pontos de comparação é o total de pontos que uma pessoa ganhou.

Dado a e b , determine seus respectivos pontos de comparação.

Por exemplo, $a = [1, 2, 3]$ e $b = [3, 2, 1]$. Por elementos, Bob recebe um ponto porque $a[0] < b[0]$. Para os elementos iguais e $a[1] = b[1]$, nenhum ponto é ganho.

Finalmente, para elementos, então Alice recebe um ponto. Sua matriz de retorno seria com a pontuação de Alice primeiro e com a segunda de Bob. $a = [1, 2, 3]$ $b = [3, 2, 1]$ $a[0] < b[0]$ $a[1] = b[1]$ $a[2] > b[2]$ $[1, 1]$

Descrição da função

Complete a função `compareTriplets` no editor abaixo. Ele deve retornar uma matriz de dois números inteiros, o primeiro sendo a pontuação de Alice e o segundo sendo o de Bob.

`compareTriplets` possui o(s) seguinte(s) parâmetro(s):

- a : uma matriz de números inteiros representando a classificação de desafio de Alice
- b : uma matriz de números inteiros representando a classificação de desafio de Bob

Formato de entrada

A primeira linha contém 3 inteiros separados por espaços, $a[0]$, $a[1]$ e $a[2]$, descrevendo os respectivos valores em triplo a . A segunda linha contém 3 inteiros separados por espaços, $b[0]$, $b[1]$ e $b[2]$, descrevendo os respectivos valores em triplo b .

Restrições

- $1 \leq a[i] \leq 100$
- $1 \leq b[i] \leq 100$

Formato de saída

Retorne uma matriz de dois números inteiros indicando os respectivos pontos de comparação ganhos por Alice e Bob.

Formato de saída

Retorne uma matriz de dois números inteiros indicando os respectivos pontos de comparação ganhos por Alice e Bob.

Entrada de amostra 0

```
5 6 7
3 6 10
```

Saída de amostra 0

```
1 1
```

Explicação 0

Neste exemplo:

- $a = (a[0], a[1], a[2]) = (5, 6, 7)$
- $b = (b[0], b[1], b[2]) = (3, 6, 10)$

Agora, vamos comparar cada pontuação individual:

- $a[0] > b[0]$, então Alice recebe um **1** ponto.
- $a[1] = b[1]$, então ninguém recebe um ponto.
- $a[2] < b[2]$, então Bob recebe **1** pontos.

A pontuação de comparação de Alice é **1** e a de comparação de Bob é **1**. Assim, retornamos a matriz $[1, 1]$

Entrada de amostra 1

```
17 28 30
99 16 8
```

Saída de amostra 1

```
2 1
```

Explicação 1

Comparando os elementos, Bob recebe um ponto. Comparando o e elementos, e então Alice recebe dois pontos. A matriz de

retorno é . $0^{th} 17 < 99$

$1^{st} 2^{nd} 28 > 16 30 > 8$

$[2, 1]$

John trabalha em uma loja de roupas. Ele tem uma grande pilha de meias que ele deve emparelhar por cores para venda. Dada uma matriz de números inteiros representando a cor de cada meia, determine quantos pares de meias com as cores correspondentes.

Por exemplo, existem meias com cores . Há um par de cores e um de cor . Restam três meias estranhas, uma de cada cor. O número de pares é . $n = 7$ $ar = [1, 2, 1, 2, 1, 3, 2]$ 122

Descrição da função

Complete a função `sockMerchant` no editor abaixo. Ele deve retornar um número inteiro representando o número de pares correspondentes de meias disponíveis.

`sockMerchant` tem o (s) seguinte (s) parâmetro (s):

- `n` : o número de meias na pilha
- `ar` : as cores de cada meia

Formato de entrada

A primeira linha contém um número inteiro n , o número de meias representado . A segunda linha contém números inteiros separados por espaço, descrevendo as cores das meias na pilha. ar

$ar[i]$

Restrições

- $1 \leq n \leq 100$
- $1 \leq ar[i] \leq 100$ Onde $0 \leq i < n$

Formato de saída

Retorne o número total de pares de meias que John pode vender.

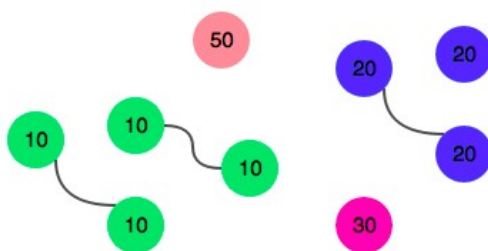
Entrada de amostra

```
9
10 20 20 10 10 30 50 10 20
```

Saída de amostra

```
3
```

Explicação



John pode combinar três pares de meias.

Gary é um ávido caminhante. Ele rastreia suas caminhadas meticulosamente, prestando muita atenção a pequenos detalhes, como topografia. Durante sua última caminhada, ele deu exatamente os n passos. Para cada passo que dava, ele notou se era uma subida, U ou uma descida, D passo. As caminhadas de Gary começam e terminam no nível do mar e cada degrau para cima ou para baixo representa uma 1 mudança de unidade na altitude. Definimos os seguintes termos:

- Uma montanha é uma sequência de etapas consecutivas acima do nível do mar, começando com uma etapa acima do nível do mar e terminando com uma etapa abaixo do nível do mar.
- Um vale é uma sequência de etapas consecutivas abaixo do nível do mar, começando com uma etapa abaixo do nível do mar e terminando com uma etapa acima do nível do mar.

Dada a sequência de degraus para cima e para baixo de Gary durante sua última caminhada, encontre e imprima o número de vales pelos quais ele atravessou.

Por exemplo, se o caminho de Gary estiver , ele primeiro entra em um vale profundo. Então ele sobe uma montanha acima.

Finalmente, ele volta ao nível do mar e termina sua caminhada. $s = [DDUUUUDD]$

Descrição da função

Complete a função `countingValleys` no editor abaixo. Ele deve retornar um número inteiro que denota o número de vales que Gary atravessou.

`countingValleys` possui o (s) seguinte (s) parâmetro (s):

- n : o número de etapas que Gary executa
- s : uma string descrevendo seu caminho

Formato de entrada

A primeira linha contém um número inteiro n , o número de etapas na caminhada de Gary.

A segunda linha contém uma única sequência s de n caracteres que descrevem seu caminho.

Restrições

- $2 \leq n \leq 10^6$
- $s[i] \in \{UD\}$

Formato de saída

Imprima um único número inteiro que denota o número de vales pelos quais Gary percorreu durante sua caminhada.

Entrada de amostra

```
8
UDDDUDUU
```

Saída de amostra

1

Explicação

Se representamos _ como nível do mar, um aumento / e uma queda como \, a caminhada de Gary pode ser desenhada como:

_/\ _
 \< /
 \\ /
 \\ /

Ele entra e sai de um vale.