

# Fita Colorida

Nome do arquivo: `fita.c`, `fita.cpp`, `fita.pas`, `fita.java`, `fita.js` ou `fita.py`

Roberto tem um conjunto de lápis com 10 tons diferentes de uma mesma cor, numerados de 0 a 9. Numa fita quadriculada, alguns quadrados foram coloridos inicialmente com o tom 0. Roberto precisa determinar, para cada quadrado  $Q$  não colorido, qual é a distância dele para o quadrado mais próximo de tom 0. A distância entre dois quadrados é definida com o número mínimo de movimentos para a esquerda, ou para a direita, para ir de um quadrado para o outro. O quadrado  $Q$ , então, deve ser colorido com o tom cuja numeração corresponde à distância determinada. Se a distância for maior ou igual a 9, o quadrado deve ser colorido com o tom 9. Seu programa deve colorir e imprimir a fita quadriculada dada na entrada.

## Entrada

A primeira linha da entrada contém apenas um inteiro  $N$ , indicando o número de quadrados da fita. A segunda linha contém  $N$  números inteiros: “-1” se o quadrado não está colorido, e “0” se está colorido com o tom 0.

## Saída

Seu programa deve escrever na saída a fita totalmente colorida, de acordo com a regra definida acima.

## Restrições

- $3 \leq N \leq 10000$ ;
- Sempre existe pelo menos um “0” inicialmente na fita.

## Informações sobre a pontuação

- Em um conjunto de casos de teste somando 80 pontos,  $N \leq 1000$

## Exemplos

<b>Entrada</b> 8 -1 -1 0 -1 -1 -1 0 -1	<b>Saída</b> 2 1 0 1 2 1 0 1
<b>Entrada</b> 13 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 0 -1 -1	<b>Saída</b> 1 0 1 2 3 4 4 3 2 1 0 1 2
<b>Entrada</b> 6 0 -1 -1 -1 -1 -1	<b>Saída</b> 0 1 2 3 4 5

# Sanduíche

Nome do arquivo: `sanduche.c`, `sanduche.cpp`, `sanduche.pas`, `sanduche.java`,  
`sanduche.js`, `sanduche.py2` ou `sanduche.py3`

Você está na Seletiva para a IOI e depois de um dia cansativo de provas, chegou a hora do jantar. Hoje, trouxeram um sanduíche muito longo cortado em  $N$  pedaços de diversos tamanhos diferentes. Você gostaria de comer uma quantidade total de sanduíche de comprimento  $D$ , porém há uma regra: para evitar bagunça, você só pode ou pegar uma sequência contínua de pedaços, ou pegar pedaços das extremidades. Você sabe a sequência  $C_1, C_2, \dots, C_N$  dos comprimentos dos pedaços na ordem em que estão posicionados no sanduíche. Agora, para otimizar o seu jantar, quer fazer um programa que com esses dados responda de quantas formas você pode escolher os pedaços do sanduíche que vai comer. Em outras palavras, deve contar quantos pares  $(i, j)$ ,  $1 \leq i \leq j \leq N$ , existem tais que o somatório  $C_i + C_{i+1} + \dots + C_j$  seja igual a  $D$  e quantos pares  $(i, j)$ ,  $1 \leq i < j \leq N$ , existem tais que o somatório  $C_1 + C_2 + \dots + C_i + C_j + C_{j+1} + \dots + C_N$  seja igual a  $D$ .

## Entrada

A primeira linha contém dois inteiros  $N$  e  $D$ , representando respectivamente o número de pedaços e a quantidade de sanduíche que você quer comer. A segunda linha contém  $N$  inteiros  $C_1, C_2, \dots, C_N$ , onde  $C_i$  é o tamanho do  $i$ -ésimo pedaço.

## Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o número de maneiras de comer pedaços de sanduíche com soma  $D$ .

## Restrições

- $2 \leq N \leq 10^6$ .
- $1 \leq D \leq 10^9$ .
- $1 \leq C_i \leq 10^3$ .

## Informações sobre a pontuação

- Em um conjunto de casos de teste equivalente a 20 pontos,  $N \leq 200$ .
- Em um conjunto de casos de teste equivalente a 40 pontos,  $N \leq 1000$ .

## Exemplos

<b>Entrada</b> 5 10 1 2 3 4 3	<b>Saída</b> 3
<b>Entrada</b> 5 5 1 1 1 1 1	<b>Saída</b> 5

Entrada	Saída
9 618 665 658 248 282 428 562 741 290 457 5	0