

**LISTA DE ALGORITMOS - VETORES**  
Prof. [Guilherme Apolinário Silva Novaes](#)

- 121) Faça um programa que receba 10 números inteiros e mostre-os de forma invertida.
- 122) Faça um programa que receba um número inteiro  $n$ , e em seguida, receba  $n$  números inteiros. Seu programa deve mostrar a quantidade de cada número digitado (O valor máximo de  $n$  é 50).
- 123) Faça um programa que receba uma sequência A e B, de 10 números reais cada, e mostre na tela a soma de cada um  $A + B$ , dos 10 números.
- 124) Faça um programa que receba 10 números reais e apresente qual é o número de valor médio (mediana).
- 125) Faça um programa que receba 10 números reais e diga qual é a soma do menor com o número médio.
- 126) Faça um programa que receba 10 números reais e diga se a média desse vetor é maior que o seu número médio.
- 127) Faça um programa que receba 10 números reais e diga a soma de todos os números abaixo da média.
- 128) Faça um programa que receba uma sequência de 10 números e diga qual é a soma dos valores acima da média.
- 129) Faça um programa que converta um número de binário para decimal.
- 130) Faça um programa que converta um número decimal para binário.
- 131) Faça um programa que receba 10 números inteiros e mostre na tela "SIM" se a média da diferença entre cada número digitado for maior que a média de todos os números digitados e "NAO" caso contrário.
- 132) Faça um programa que receba 10 números inteiros e mostre na tela a quantidade de números ímpares.
- 133) Faça um programa que receba 10 números inteiros e mostre na tela o maior e o segundo menor número dentre eles.
- 134) Faça um programa que receba 10 números reais e mostre na tela o valor da maior frequência de números dentre eles. (Ex: [3,2,3,1,4,5,3,7,3,6], saída = 4)
- 135) Faça um programa que receba 10 números reais e mostre na tela o número de maior frequência dentre eles.
- 136) Faça um programa que receba 10 números reais e mostre na tela quantos números são maiores que o quinto número digitado.
- 137) Faça um programa que receba 10 números reais e mostre na tela a soma dos três primeiros números multiplicados pelo sexto número e por fim, subtraia os demais números.

- 138) Faça um programa que receba dois vetores, A e B, de 10 números cada, e apresente na tela quantos números em comum A e B possuem.
- 139) Faça um programa que receba 10 números inteiros, e mostre na tela a soma dos cinco primeiros com a subtração dos 5 últimos.
- 140) Faça um programa que receba 10 números reais e mostre na tela, de acordo com a ordem que os números foram digitados, o dobro de cada número na posição ímpar, e a metade de cada número da posição par (o vetor começa na posição 1).
- 141) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Se a soma total dos números for par, mostre na tela o dobro de todos os números digitados, caso contrário, mostre na tela a metade destes números.
- 142) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre na tela todos os números na ordem que foram digitados, e em seguida, mostre todos os números na ordem oposta que foram digitados.
- 143) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre apenas os números que foram digitados nas posições ímpares (Ex: 1°, 3°, 5°, ...).
- 144) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre na tela apenas os números que foram digitados nas posições pares, mas de trás pra frente.
- 145) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre na tela os primeiros  $n/2$  números na ordem que foram digitados, e os últimos  $n/2$  números na ordem contrária que foram digitados.
- 146) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre na tela todos os números que estão entre o terceiro e o sexto número digitados.
- 147) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre na tela todos os números entre o primeiro e o segundo (incluindo o primeiro e o segundo), o segundo e o terceiro, terceiro e quarto, até o  $n-1$  e  $n$ .
- 148) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre na tela a diferença entre a soma de todos os valores nas posições pares, com a soma de todos os valores nas posições ímpares.
- 149) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre na tela a quantidade de números pares entre os  $n/2$  primeiros números, e a quantidade de números ímpares entre os  $n/2$  últimos números.

- 150) Faça um programa que receba um número inteiro  $n$  e, em seguida, receba  $n$  números inteiros. Mostre os números nas posições ímpares na ordem que foram digitados, e os números das posições pares na ordem reversa que foram digitados. (Ex: 1,2,3,4,5,6 -> 1,6,3,4,5,2).
- 151) Faça um programa que receba 10 números inteiros e apresente na tela o resultado da multiplicação de todos esses números dividido pelo valor do sétimo número.
- 152) Faça um programa que receba 10 números reais, e logo em seguida, receba mais dois números inteiros,  $a$  e  $b$ . Mostre na tela todos os números na ordem digitado, com exceção dos números das posições  $a$  e  $b$ , que devem aparecer trocados. Caso  $a$  ou  $b$  forem maiores que 10 ou menores que 0, o programa deverá exibir "Impossível" na tela.
- 153) Faça um programa de gerenciamento de estacionamento. O estacionamento possui 10 vagas. Toda vez que um veículo entrar no estacionamento, caso haja alguma vaga livre, o veículo estaciona naquela vaga. A entrada do programa deve ser um número inteiro indicando a quantidade de carros que entram no estacionamento, até que seja digitado 0. Após digitar os números, o programa deverá parar de receber entradas e deverá mostrar quantos lugares vagos há no estacionamento. (Dica: Utilize um vetor do tipo lógico).
- 154) Faça um programa para gerenciar uma lista de compras. Cada posição do vetor equivale a um produto, sendo: 1) Abacaxi - R\$10,00, 2) Cenoura - R\$ 3,00, 3) Abacate - R\$ 4,50, 4) Beterraba R\$ 9,40, 5) Alface - R\$ 6,00, 6) Carne - R\$ 29,90, 7) Feijão - R\$ 9,40, 8) Arroz - R\$ 10,00, 9) Ameixa - R\$ 10,00, 10) Rúcula - R\$ 7,00. O programa deverá receber como entrada 10 números inteiros, indicando a quantidade de itens comprados, e deverá exibir o valor total da compra após ser digitado todos os valores.
- 155) Faça um programa para verificar se um CPF é válido, baseado em seu [dígito verificador](#).
- 156) Faça um programa que receba 10 números inteiros e exiba aleatoriamente dois deles na tela.
- 157) Faça um programa que receba 10 números inteiros e exiba na tela a seguinte sequência: soma dos 5 primeiros números; soma do número da posição 1, até a posição 6; da posição 2, até a 7, e assim até a posição 10. Ex: [0,1,2,3,4,5,6,7,8,9] => 10, 16, 20, 25, ...
- 158) Faça um programa que receba dois números inteiros e apresente na tela o quadrado dos 100 primeiros números da seguinte sequência: soma dos 2 primeiros números; soma do segundo com o terceiro número; soma do terceiro, com o quarto; soma do quarto, com o quinto; e assim sucessivamente.

- 159) Faça um algoritmo que receba 11 números inteiros e mostre na tela quantos desses números são primos, quantos são pares e quantos são ímpares.
- 160) É sabido que o nono dígito de um CPF representa o estado do qual o documento foi emitido (Exemplo: 147.868.920-09 é um CPF do Rio Grande do Sul). Sendo assim, faça um programa que receba todos os 11 números inteiros de um CPF e apresente na tela o seguinte padrão:

9º Dígito	Saída do programa
0	Rio Grande do Sul
1	Distrito Federal, Goiás, Mato Grosso, Mato Grosso do Sul ou Tocantins
2	Amazonas, Pará, Roraima, Amapá, Acre ou Rondônia
3	Ceará, Maranhão ou Piauí
4	Paraíba, Pernambuco, Alagoas ou Rio Grande do Norte
5	Bahia ou Sergipe
6	Minas Gerais
7	Rio de Janeiro ou Espírito Santo
8	São Paulo
9	Paraná ou Santa Catarina

- 161) Faça um programa que receba  $n$  números inteiros até que seja digitado 0. A cada número que for digitado, seu programa deverá exibir na tela, os 5 maiores números, diferente de zero, digitados até então.
- 162) Faça um programa que receba 10 números inteiros e mostre na tela todos os números que só foram digitados uma única vez.
- 163) Faça um programa que receba 10 números inteiros e mostre na tela todos os números que só foram digitados mais de uma vez.
- 164) Faça um programa que receba 10 números inteiros e mostre na tela o seu valor negativo, caso seja um número par, e o dobro de seu valor, caso seja um número ímpar.
- 165) Faça um programa que receba 10 números inteiros, e depois receba a entrada de um único número inteiro. Seu programa deverá mostrar todos os 10 primeiros números digitados, com exceção do último número digitado.

- 166) Faça um programa que receba 20 números inteiros, divididos em dois vetores de tamanho 10. Exiba na tela todos os elementos que os 2 vetores não possuem em comum.
- 167) Faça um programa que receba 10 números inteiros, divididos em dois vetores de tamanho 5. Exiba na tela todos os elementos que os 2 vetores possuem em comum.
- 168) Faça um programa que receba 10 números inteiros, em seguida, um número inteiro. Exiba na tela o índice do último valor digitado no vetor. Caso o valor digitado não esteja contido no vetor, exiba -1.
- 169) Faça um programa que receba 10 números inteiros, e guarde-os em dois vetores, A e B, de capacidade 5. Exiba na tela a seguinte ordem: 1º elemento de A, 1º elemento de B, 2º elemento de A, 2º elemento de B, ... até o elemento 5 do vetor B.
- 170) Faça um programa que receba 10 números inteiros e exiba na tela, na ordem digitada, todos os números ímpares e depois todos os números pares.
- 171) Faça um programa que receba um número real de até 5 casas, e exiba este número ao contrário, sem a vírgula. Ex: 12.345 => 54321
- 172) Faça um programa que receba 10 números inteiros, seguido de um número inteiro. Exiba na tela todos os 10 primeiros números, mas substituindo o número informado por último pela sua metade.
- 173) Faça um programa que receba 10 números inteiros e exiba na tela o segundo maior valor, subtraído do segundo menor.
- 174) Faça um programa que receba 10 números inteiros, seguido de um número inteiro. Exiba na tela quantos dos 10 primeiros números são divisíveis pelo último número digitado.
- 175) Faça um programa que receba 10 números inteiros, divididos em 2 vetores, A e B, de tamanho 5. Exiba na tela os 2 primeiros elementos do vetor A, seguido dos 3 últimos do vetor B. Faça o mesmo para os 2 primeiros de B e os 3 últimos de A.
- 176) Faça um programa que receba 10 números inteiros e exiba na tela "SIM" se o terceiro número for maior que a média. Exiba "NAO" caso contrário.
- 177) Faça um programa que receba 10 números inteiros, seguindo de um número  $n$  inteiro. Exiba na tela todos os números de 0 até  $n$ , com exceção dos 10 primeiros números digitados.
- 178) Faça um programa que receba 10 números inteiros, e guarde-os em dois vetores, A e B, de capacidade 5. Exiba na tela todos os valores que não se repetem, apenas do vetor A.
- 179) Faça um programa que receba 10 números inteiros, e guarde-os em dois vetores, A e B, de capacidade 5. Exiba na tela a distância entre esses dois vetores. (Dica: A fórmula é a mesma da fórmula da distância entre dois pontos)

- 180) Faça um programa que receba 20 números reais, divididos em dois vetores de capacidade 10, e calcule seu produto escalar.
- 181) Faça um programa que receba 10 números reais e calcule o desvio padrão deles. A fórmula do desvio padrão é ( $\mu$  = Média

$$= \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2},$$

dos elementos) =

- 182) Faça um programa que receba 10 números reais e mostre na tela qual é a diferença entre a mediana e a média desses números.
- 183) Faça um programa que receba 10 números inteiros e mostre na tela o MMC deles.
- 184) Faça um programa que receba 10 números inteiros e mostre na tela qual é a sua [moda](#).
- 185) Faça um programa que receba 10 números reais e mostre na tela a raiz cúbica da soma de todos os seus elementos.
- 186) Faça um programa que receba 5 números inteiros entre 0 e 9, e mostre na tela a raiz quadrada da sequência de números digitados. Ex: 1 2 3 4 5 =  $\sqrt{12345} \approx 111,108056$
- 187) Faça um programa que receba 5 números inteiros entre 0 e 9, e apresente na tela a soma de sua sequência com o seu inverso. Ex: 1 2 3 4 5 = 12345+54321 = 66666
- 188) Faça um programa que receba 6 números inteiros entre 0 e 9, e some suas duplas. Ex: 1 2 3 4 5 6 = 12 + 34 + 56 = 102
- 189) Faça um programa que receba dois números inteiros, A e B, e apresente os 5 últimos restos da divisão de A e B.
- 190) Faça um programa que receba 10 números inteiros e mostre-os de forma aleatória.
- 191) Faça um programa que receba 10 números reais, some-os, e mostre na tela se a sua soma possui algum número em suas casas decimais.
- 192) Faça um programa que receba um número inteiro e apresente na tela "SIM", se ele pode ser representado através de uma soma de sequência de números (Ex: 1+2+3+4 = 10), seguido dos números que somam, em ordem decrescente. Mostre "NAO", caso contrário.
- 193) Faça um programa que receba 10 números inteiros e mostre-os de ordem crescente.
- 194) Faça um programa que receba 10 números inteiros e mostre-os de ordem decrescente.
- 195) Faça um programa que receba 10 números reais, 5 em cada um de dois vetores, e apresente na tela a similaridade de cosseno entre eles. A fórmula é descrita por:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

196) [Desafio] Faça um programa que receba dois vetores de números reais, A e B, de 5 números, e exiba na tela a correlação de A e B.

197) [Desafio] (1ª Fase da Maratona de Programação 2013) O recente terremoto em Nlogônia não chegou a afetar muito as edificações da capital, principal epicentro do abalo. Mas os cientistas detectaram que o principal dique de contenção teve um dano significativo na sua parte subterrânea que, se não for consertado rapidamente, pode causar o seu desmoronamento, com a consequente inundação de toda a capital.

O conserto deve ser feito por mergulhadores, a uma grande profundidade, em condições extremamente difíceis e perigosas. Mas como é a sobrevivência própria cidade que está a em jogo, seus moradores acudiram em grande número como voluntários para essa perigosa missão.

Como é tradicional em missões perigosas, cada mergulhador recebeu no início do mergulho uma pequena placa com um número de identificação. Ao terminar o mergulho, os voluntários devolviam a placa de identificação, colocando-a em um repositório.

O dique voltou a ser seguro, mas aparentemente alguns voluntários não voltaram do mergulho. Você foi contratado para a penosa tarefa de, dadas as placas colocadas no repositório, determinar quais voluntários perderam a vida salvando a cidade.

### Entrada

A entrada é composta de duas linhas. A primeira linha contém dois inteiros  $N$  e  $R$ , indicando respectivamente o número de voluntários que mergulhou e o número de voluntários que retornou do mergulho. Os voluntários são identificados por números de 1 a  $N$ . A segunda linha da entrada contém  $R$  inteiros, indicando os voluntários que retornaram do mergulho (ao menos um voluntário retorna do mergulho).

### Saída

Seu programa deve produzir uma única linha, contendo os identificadores dos voluntários que não retornaram do mergulho, na ordem crescente de suas identificações. Deixe um espaço em branco após cada identificador (note que isto significa que deve haver um espaço em branco também após o último identificador). Se todos os voluntários retornaram do mergulho, imprima apenas o caractere '\*' (asterisco).

### Restrições

$$1 \leq R \leq N \leq 10^4$$

### Exemplos

<b>Entrada</b> 5 3 3 1 5	<b>Saída</b> 2 4
<b>Entrada</b> 6 6 6 1 3 2 5 4	<b>Saída</b> *



198) [Desafio] (1ª Fase da Maratona de Programação 2015) Um jogo de estratégia, com  $J$  jogadores, é jogado em volta de uma mesa. O primeiro a jogar é o jogador 1, o segundo a jogar é o jogador 2 e assim por diante. Uma vez completada uma rodada, novamente o jogador 1 faz sua jogada e a ordem dos jogadores se repete novamente. A cada jogada, um jogador garante uma certa quantidade de Pontos de Vitória. A pontuação de cada jogador consiste na soma dos Pontos de Vitória de cada uma das suas jogadas.

Dado o número de jogadores, o número de rodadas e uma lista representando os Pontos de Vitória na ordem em que foram obtidos, você deve determinar qual é o jogador vencedor. Caso mais de um jogador obtenha a pontuação máxima, o jogador com pontuação máxima que tiver jogado por último é o vencedor.

### Entrada

A entrada consiste de duas linhas. A primeira linha contém dois inteiros  $J$  e  $R$ , o número de jogadores e de rodadas respectivamente ( $1 \leq J, R \leq 500$ ). A segunda linha contém  $J \times R$  inteiros, correspondentes aos Pontos de Vitória em cada uma das jogadas feitas, na ordem em que aconteceram. Os Pontos de Vitória obtidos em cada jogada serão sempre inteiros entre 0 e 100, inclusive.

### Saída

Seu programa deve produzir uma única linha, contendo o inteiro correspondente ao jogador vencedor.

### Exemplos

<b>Entrada</b> 3 3 1 1 1 1 2 2 2 3 3	<b>Saída</b> 3
<b>Entrada</b> 2 3 0 0 1 0 2 0	<b>Saída</b> 1

199) [Desafio] (1ª fase da Maratona de Programação 2011) A divisão de Suprimentos de Botas e Calçados do Exército comprou um grande número de pares de botas de vários tamanhos para seus soldados. No entanto, por uma falha de empacotamento da fábrica contratada, nem todas as caixas entregues continham um par de botas correto, com duas botas do mesmo tamanho, uma para cada pé. O sargento mandou que os recrutas retirassem todas as botas de todas as caixas para reembalá-las, desta vez corretamente.

Quando o sargento descobriu que você sabia programar, ele solicitou com a gentileza habitual que você escrevesse um programa que, dada a lista contendo a descrição de cada bota entregue, determina quantos pares corretos de botas poderão ser formados no total.

### Entrada

A primeira linha de um caso de teste contém um inteiro  $N$  indicando o número de botas individuais entregues. Cada uma das  $N$  linhas seguintes descreve uma bota, contendo um número inteiro  $M$  e uma letra  $L$ , separados por um espaço em branco.  $M$  indica o número do tamanho da bota e  $L$  indica o pé da bota:  $L = 'D'$  indica que a bota é para o pé direito,  $L = 'E'$  indica que a bota é para o pé esquerdo.

### Saída

Para cada caso de teste imprima uma linha contendo um único número inteiro indicando o número total de pares corretos de botas que podem ser formados.

### Restrições

- $2 \leq N \leq 10^4$
- $N$  é par
- $30 \leq M \leq 60$
- $L \in \{D, E\}$

### Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
4 40 D 41 E 41 D 40 E	2

Exemplo de entrada	Saída para o exemplo de entrada
6 38 E 39 E 40 D 38 D 40 D 37 E	1

200) [Desafio] (URI Online Judge | Picos e Vales) Ao observar a paisagem da Nlogônia, o professor MC percebeu que a cada intervalo de 100 metros existe um pico. E que exatamente na metade de dois picos há um vale. Logo, a cada 50 metros há um vale ou um pico e, ao longo da paisagem, não há um pico seguido por outro pico, nem um vale seguido por outro vale.

O professor MC ficou curioso com esse padrão e quer saber se, ao medir outras paisagens, isso se repete. Sua tarefa é, dada uma paisagem, indicar se ela possui esse padrão ou não.

### Entrada

A entrada é dada em duas linhas. A primeira tem o número  $N$  de medidas da paisagem ( $1 < N \leq 100$ ). A segunda linha tem  $N$  inteiros: a altura  $H_i$  de cada medida ( $-10000 \leq H_i \leq 10000$ , para todo  $H_i$ , tal que  $1 \leq i \leq N$ ). Uma medida é considerada um pico se é maior que a medida anterior. Uma medida é considerada um vale se é menor que a medida anterior.

### Saída

A saída é dada em uma única linha. Caso a paisagem tenha o mesmo padrão da Nlogônia, deve ser mostrado o número 1. Caso contrário, mostra-se o número 0.

### Exemplos

<b>Entrada</b> 3 1 4 -2	<b>Saída</b> 1
<b>Entrada</b> 5 100 99 112 -8 -7	<b>Saída</b> 1
<b>Entrada</b> 4 1 2 2 1	<b>Saída</b> 0