

Condicionais e Loops

Técnicas de Programação - 2025/02

Última atualização: 13/08/2025 11:22

Para cada algoritmo abaixo:

1. leia o pseudo código
2. simule seu funcionamento usando as entradas de exemplo
3. confira o resultado usando os testes no exercício do PrairieLearn
4. use o *CheatSheet* para converter o pseudo código para Java

Entregador mais próximo

Algorithm 1: Entregador mais próximo

Result: Encontra o índice (x_i, y_i) mais perto do ponto o

Input : array float x , array float y , float o_x , float o_y

Output: int

int $mprox \leftarrow 0$

float $dist_mprox \leftarrow \sqrt{(x_0 - o_x)^2 + (y_0 - o_y)^2}$

for $i \leftarrow 1$ **to** $x.length$ **do**

float $d = \sqrt{(x_i - o_x)^2 + (y_i - o_y)^2}$

if $d < dist_mprox$ **then**

$dist_mprox \leftarrow d$

$mprox \leftarrow i$

end

end

return $mprox$

Entrada 1: $o = (0, 0), x = [123, 42, 10], y = [432, 312, 20]$

Entrada 2: $o = (3, 4), x = [10, 4, 4, 23], y = [20, 2, 4, 63]$

Diferença de listas

Algorithm 2: DiffListas

Result: Cria nova lista com diferença entre A_1 e A_2

Input : array A_1 , array A_2

Output: array A_3

```

count ← 0
for i ← 0 to A1.length do
    emA2 ← false
    for f ← 0 to A2.length do
        if A1[i] = A2[j] then
            | emA2 ← true
        end
    end
    if emA2 = false then
        | count ← count + 1
    end
end
A3 = novo array(count)
count ← 0
for i ← 0 to A1.length do
    emA2 ← false
    for f ← 0 to A2.length do
        if A1[i] = A2[j] then
            | emA2 ← true
        end
    end
    if emA2 = false then
        | A3[count] = A1[i]
        | count ← count + 1
    end
end
return A3

```

Entrada 1: $A_2 = [1, 2, 3, 4]$, $A_1 = [4, 5, 6]$

Entrada 2: $A_1 = [1, 2, 3, 4]$, $A_2 = [4, 5, 6]$

Valor da nota

Algorithm 3: ValorNota

Result: Valor da Nota**Input** : array float P , array float Q **Output:** float $total \leftarrow 0$ **for** $i \leftarrow 0$ **to** $P.length$ **do**| $total \leftarrow total + P[i] * Q[i]$ **end****return** $total$

Entrada 1: $P = [1.40, 1.45, 11.4]$, $Q = [1.40, 1.45, 11.4]$

Entrada 2: $P = [1.5]$, $Q = [3]$

Entrada 3: $P = [1.21, 1.1, 4.5, 6.1, 100.15, 0.4, 8.67]$, $Q = [13, 41, 63, 6, 52, 14, 47]$

Conta Celulares

Algorithm 4: FiltraCelulares

Result: Dada uma lista com números de telefone, devolve uma nova lista contendo somente os números que são de celular.

Input : array string L

Output: array string

```

count  $\leftarrow$  0; for  $i \leftarrow 0$  to  $L.length$  do
    if  $L[i].length = 14$  AND  $L[i][5] = '9'$  then
        | count  $\leftarrow$  count + 1
    end
    if  $L[i].length = 11$  AND  $L[i][2] = '9'$  then
        | count  $\leftarrow$  count + 1
    end
    if  $L[i].length = 9$  AND  $L[i][0] = '9'$  then
        | count  $\leftarrow$  count + 1
    end
end
res  $\leftarrow$  novo array string[count]
k  $\leftarrow$  0
for  $i \leftarrow 0$  to  $L.length$  do
    if  $L[i].length = 14$  AND  $L[i][5] = '9'$  then
        | res[k]  $\leftarrow$  L[i]
        | k  $\leftarrow$  k + 1
    end
    if  $L[i].length = 11$  AND  $L[i][2] = '9'$  then
        | res[k]  $\leftarrow$  L[i]
        | k  $\leftarrow$  k + 1
    end
    if  $L[i].length = 9$  AND  $L[i][0] = '9'$  then
        | res[k]  $\leftarrow$  L[i]
        | k  $\leftarrow$  k + 1
    end
end
return res

```

Entrada 1: $L = [“+5511912345678”, “1155556666”, “77778888”, “+55113334444”, “918273645”, “11987654321”]$

Palavras iguais

Agora vamos tentar escrever um pseudo código nosso. Não se esqueça que nossas strings são acessadas igual a um array. Preencha abaixo e verifique se seu algoritmo funciona usando as entradas abaixo.

Algorithm 5: PalavrasIguais

Result: Verifica se uma string é formada por duas palavras iguais separadas por um caractere '-'

Input : string S

Output: boolean

Entrada 1: $S = \text{"abc-abc"}$

Entrada 2: $S = \text{"abc-abca"}$

Entrada 3: $S = \text{"zig-zag"}$