

# LISTAS E REPETIÇÃO. SOLUÇÕES AOS EXERCÍCIOS

MC102 - Algoritmos e  
Programação de  
Computadores

Santiago Valdés Ravelo  
[https://ic.unicamp.br/~santiago/  
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

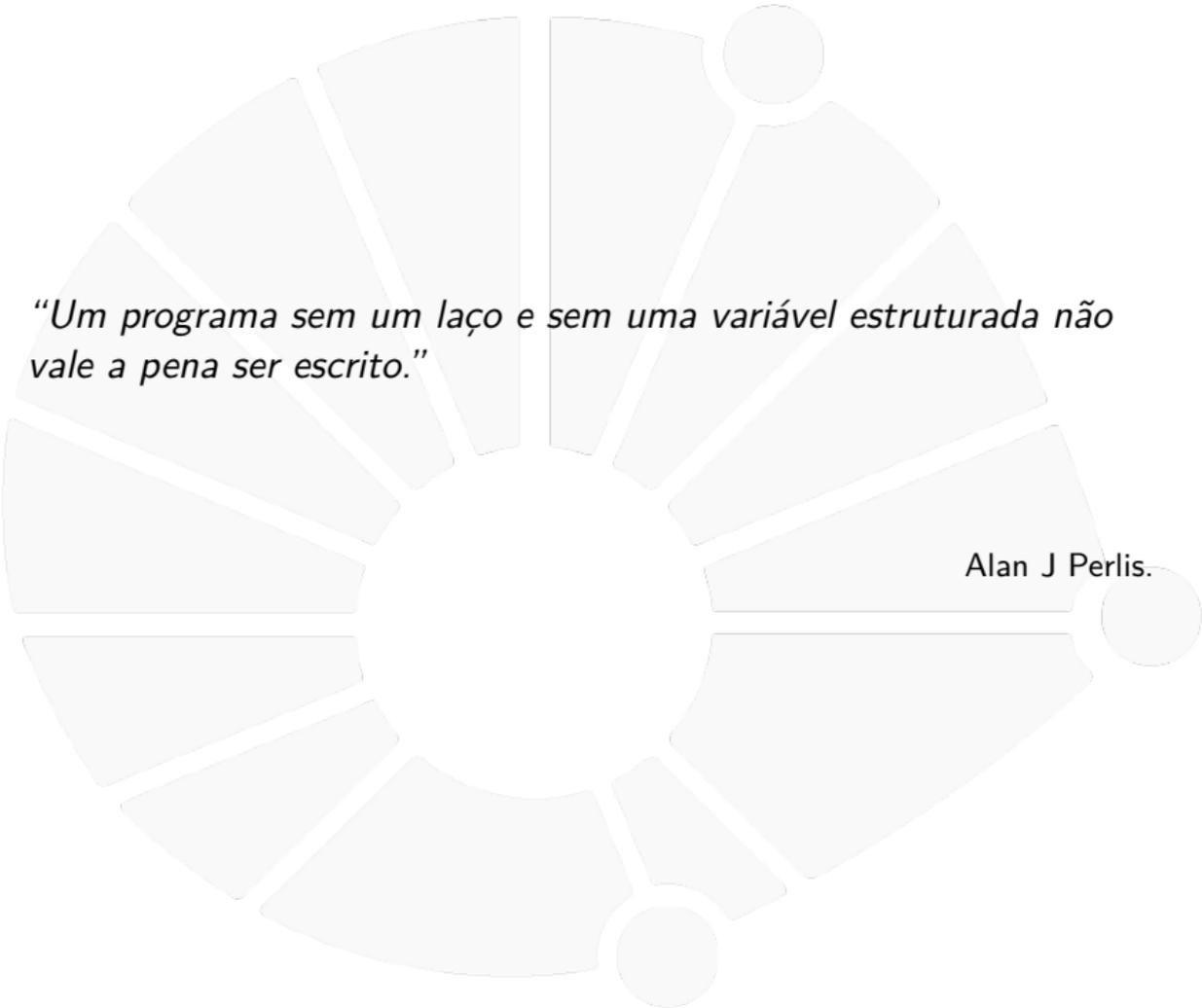
03/25

6



UNICAMP



A decorative circular graphic composed of light gray segments and circles. The graphic is a large circle divided into 16 equal segments by white lines. At the top, bottom, left, and right positions, one of these segments is replaced by a solid light gray circle. The quote is centered within the circle.

*“Um programa sem um laço e sem uma variável estruturada não vale a pena ser escrito.”*

Alan J Perlis.



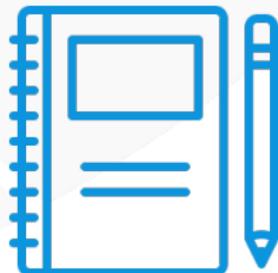
# EXERCÍCIOS

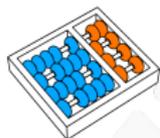


## Listas e repetição



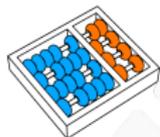
**Soluções para os exercícios!**





## Exercício 1

- (a) Dado  $n$ , imprima todos os números ímpares menores do que  $n$ .
- (b) Imprima os  $n$  primeiros números naturais em ordem inversa.
- (c) Dado um inteiro  $n$  seguido por  $n$  inteiros, imprima o menor valor e o maior valor dos  $n$  inteiros, seguido pela quantidade de vezes que esses valores se repetem.

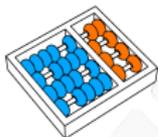


## Solução (a)

Dado  $n$ , imprima todos os números ímpares menores do que  $n$ :

```
1 n = int(input("Entre com o n: "))
2
3 for i in range(1, n, 2):
4     print(i)
```

O `range(1, n, 2)` nos dá os números menores que  $n$ , começando pelo 1 e somando de 2 em 2 (ou seja os ímpares menores que  $n$ ).



## Solução (b)

Dado  $n$ , imprima os  $n$  primeiros números naturais em ordem inversa.

```
1 n = int(input("Entre com o n: "))
2
3 for i in range(n, 0, -1):
4     print(i)
```

O `range(n, 0, -1)` nos dá os números começando por  $n$  e diminuindo de um em um até chegar no 0 (sem incluir o próprio zero, se desejarmos incluir o zero, precisamos chegar até o -1).



## Solução (c)

```
1 n = int(input("Entre com o n: "))
2 lista = [0] * n
3 i = 0
4 pos_min = 0
5 qtd_min = 0
6 pos_max = 0
7 qtd_max = 0
8 for i in range(n):
9     lista[i] = int(input())
10    if lista[pos_min] < lista[i]:
11        pos_min = i
12        qtd_min = 1
13    elif lista[pos_min] == lista[i]:
14        qtd_min += 1
15    if lista[pos_max] > lista[i]:
16        pos_max = i
17        qtd_max = 1
18    elif lista[pos_max] == lista[i]:
19        qtd_max += 1
20 print('Menor:', lista[pos_min], 'quantidade de vezes que aparece:' qtd_min)
21 print('Maior:', lista[pos_max], 'quantidade de vezes que aparece:' qtd_max)
```

Armazenamos as posições em que encontramos o menor e maior (pos\_min e pos\_max), ademais disso o número de vezes que os encontramos. Inicialmente, supomos que ambos estão na primeira posição. A medida que encontramos um novo elemento, verificamos se precisamos atualizar o menor ou o maior, em caso de atualizar é porque esta foi a primeira vez que esse valor apareceu e portanto a quantidade fica 1. Caso não atualizemos, verificamos se o valor lido é igual ao menor ou o maior, se for, então precisamos aumentar a quantidade de vezes que eles apareceram em 1.

# LISTAS E REPETIÇÃO. SOLUÇÕES AOS EXERCÍCIOS

MC102 - Algoritmos e  
Programação de  
Computadores

Santiago Valdés Ravelo  
[https://ic.unicamp.br/~santiago/  
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

03/25

6



UNICAMP

