

LISTAS E REPETIÇÃO. DE NOVO?! 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)

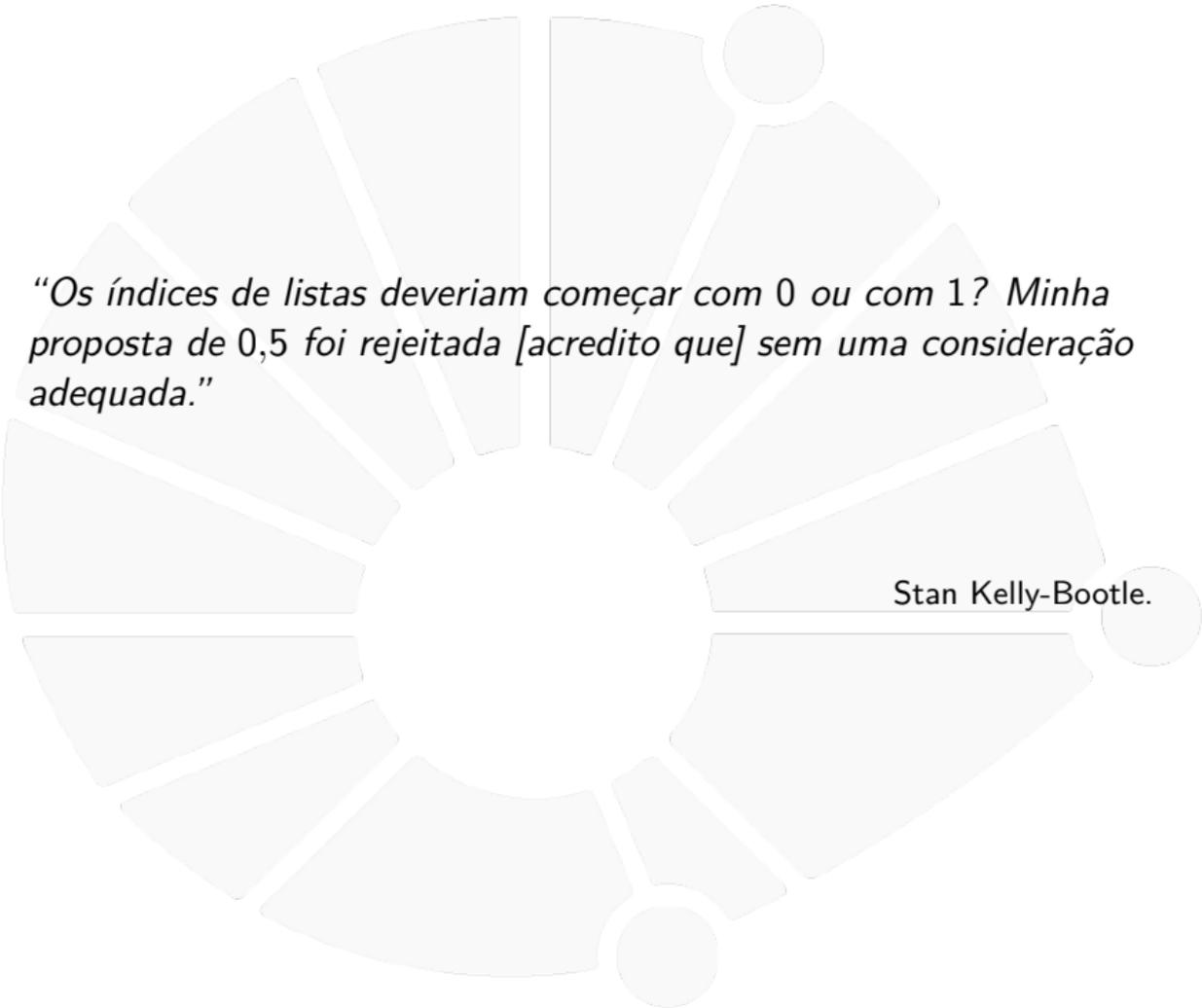
03/25

7



UNICAMP





“Os índices de listas deveriam começar com 0 ou com 1? Minha proposta de 0,5 foi rejeitada [acredito que] sem uma consideração adequada.”

Stan Kelly-Bootle.



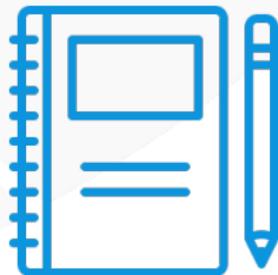
EXERCÍCIOS

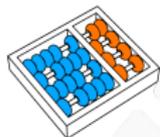


Listas e repetição



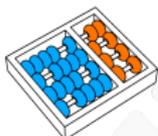
Soluções para os exercícios!





Exercícios

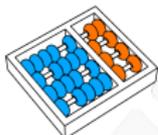
1. Faça um programa que dado uma lista de números, testa se todos são primos.
2. Faça um programa que conta quantos primos menores do que n existem.
3. Faça um programa que, dado n , imprime um quadrado formado por $*$'s com n linhas e n colunas.
 - ▶ Repita para retângulo.
 - ▶ Repita para triângulo retângulo isósceles.
4. Um número é perfeito se é igual a soma dos seus divisores (ex: $6 = 1 + 2 + 3$). Faça um programa que dado um número n , imprime todos os números perfeitos menores ou iguais a n .



Solução 1.

```
1 lista = list(map(int, input().split()))
2 # Procura o maior
3 maior = lista[0]
4 for n in lista:
5     if n > maior:
6         maior = n
7 # Usa o crivo o no maior
8 eh_primo = [True] * (maior + 1)
9 eh_primo[0:2] = [False] * 2
10 for p in range(2, maior + 1):
11     if eh_primo[p]:
12         print(p)
13         eh_primo[p * p::p] = [False] * len(eh_primo[p * p::p])
14 # Testa se todos são primos
15 todos_primos = True
16 for n in lista:
17     if not eh_primo[n]:
18         todos_primos = False
19 print(todos_primos)
```

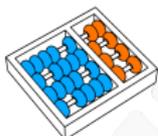
Encontra o maior elemento da lista e aplica o crivo para ele. Depois é só testar quais elementos ficaram marcados como primos.



Solução 2.

```
1 n = int(input())
2 # Usa o crivo
3 eh_primo = [True] * (n)
4 eh_primo[0:2] = [False] * 2
5 for p in range(2, n):
6     if eh_primo[p]:
7         print(p)
8         eh_primo[p * p::p] = [False] * len(eh_primo[p * p::p])
9 # Conta os primos
10 primos = 0
11 for v in eh_primo:
12     if v:
13         primos += 1
14 print(primos)
```

Aplica o crivo para $n-1$. Depois é só contar quantos elementos ficaram marcados como primos.



Solução 3.

Quadrado. Cria a linha do tamanho certo (uma string de n *'s) e imprime essa linha n vezes.

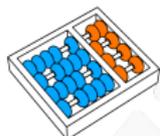
```
1 n = int(input())
2 linha = '*' * n
3 for i in range(n):
4     print(linha)
```

Retângulo. Cria a linha do tamanho certo (uma string de n *'s) e imprime essa linha m vezes.

```
1 n = int(input())
2 m = int(input())
3 linha = '*' * n
4 for i in range(m):
5     print(linha)
```

Triângulo. A primeira linha tem um *, a segunda dois, e assim até chegar em n.

```
1 n = int(input())
2 for i in range(1, n + 1):
3     print('*' * i)
```



Solução 4.

```
1 n = int(input())
2
3 soma = 1
4 k = 2
5 while k * k < n:
6     if n % k == 0:
7         soma += k + (n / k)
8
9     if n == k * k:
10        soma += k
11
12 print(soma == n)
```

Parecida à solução para ver se um número é primo, mas ao invés de parar, testamos todos os números até a raiz. Por cada divisor encontrado, somamos ele e também somamos o número que multiplicado com ele dá n (dessa forma não precisamos testar após a raiz). O caso especial é a própria raiz, que só precisa ser somada uma vez (por isso no while paramos antes de chegar nela). Finalmente o número é perfeito se for igual à soma dos seus divisores. Note que a soma começa com 1, porque 1 é divisor de todos os números.

LISTAS E REPETIÇÃO. DE NOVO?! SOLUÇÕES AOS EXERCÍCIOS

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

MC102 - Algoritmos e
Programação de
Computadores

03/25

7



UNICAMP

