

MC-102 — Aula 08

Comandos Repetitivos

Instituto de Computação – Unicamp

28 de Agosto de 2019

Primos

- da aula passada: escrever uma função que determina se um número é primo ou não. **True** se for.
- Usando aquela função, imprima os n primeiros números primos > 2

Primos

```
def primo?(x):  
    for i in range(2,x):  
        if x % i == 0:  
            return False  
    return True
```

- O **range(2,n)** para no $x - 1$. Dá para melhorar esse valor final do range para que ele gere menos números!
- por exemplo, não há divisores de x entre $x/2$ e $x - 1$. Portanto dá para parar o **range** perto/no $x/2$.
- Existe um limite ainda mais baixo? Pense no \sqrt{x} .

Primos

- O problema para ser resolvido pede os n primeiros números primos.
- é so gerar vários números seguidos (mas quantos?), chamar o **primo?** para cada um deles e imprimir os que retornam **True**
- mas quantos números precisamos gerar??
- loops que vão repetir um número não sabido de vezes (na hora de começar o loop) precisam do **while**
- ai é só contar quantos números primos foram impressos, e parar quando esse número chegar a n

Primos

```
n=int(input('Quantos primos?: '))
impressos = 0
x=2
while impressos <= n:
    if primo(x):
        print(x,'é primo')
        impressos = impressos+1
    x = x + 1
```

- da para melhorar um pouco esse programa. Os primos (a não ser o 2) tem diferença de no mínimo 2, já que todo primo tem que ser ímpar!!

Maior média

- Escreva uma função que recebe uma lista no formato:

```
[ [ nome1, [nota11, nota12, nota13..., nota 119]],  
  [ nome2, [nota21, nota22, nota23..., nota 28]],  
  ...  
  [ nome15, [nota151, nota152, nota153..., nota 1524]] ]
```

- Ou seja uma lista de pares de nomes e uma lista (de tamanho qualquer de notas)
- retorne o nome da pessoa com maior média

Maior média

- Escreva uma função que recebe uma lista no formato:

```
[ [ nome1, [nota11, nota12, nota13..., nota 119]],  
  [ nome2, [nota21, nota22, nota23..., nota 28]],  
  ...  
  [ nome15, [nota151, nota152, nota153..., nota 1524]] ]
```

- Ou seja uma lista de pares de nomes e uma lista (de tamanho qualquer de notas)
- retorne o nome da pessoa com maior média

Maior média

- O problema parece difícil. Vamos começar por resolver um problema mais fácil.
- vamos assumir que so há uma nota, e a lista é da forma

```
[ [ nome1, nota1],  
  [ nome2, nota2],  
  ...  
  [ nome15, nota15] ]
```

Maior média

```
def maiornota(l):  
    ma = 0  
    for par in l:  
        nome = par[0]  
        nota = par[1]  
        if nota > ma:  
            ma = nota  
            nomema = nome  
    return nomema
```

Maior média

- Essa função é parecida com a que descobre o posição do maior numa lista, ou o que encontra o nome da pessoa com maior nota em 2 listas
- mas agora é so uma lista, que contem pares nome-nota
- voltado ao problema original. Eu nao tenho uma so nota, eu tenho varias notas que eu preciso tirar a media
- **ESCREVA UMA FUNÇÃO** para isso

Maior média

```
def maiornota(l):  
    ma = 0  
    for par in l:  
        nome = par[0]  
        nota = media(par[1]) #####  
        if nota > ma:  
            ma = nota  
            nomema = nome  
    return nomema
```

Maior média

```
def media(l):  
    s = 0.0  
    for x in l:  
        s = s+x  
    return s/len(l)
```

Use funções

- use funções para resolver o miolo de um problema complexo
- escreva a casca externa assumindo que voce tem uma função que resolve o miolo
- e depois escreva a função

Exercício

- Faça um programa que leia um número n e imprima n linhas na tela com o seguinte formato (exemplo se $n = 6$):

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

Use funções

```
n = int(input("n?: "))
for i in range(n):
    imprimelinha(i)

def imprimelinha(i):
    for j in range(1,i+1):
        print(j, end=' ')
    print()
```

Exercício

- Faça um programa que leia um número n e imprima n linhas na tela com o seguinte formato (exemplo se $n = 6$):

```
+ * * * * *
* + * * * *
* * + * * *
* * * + * *
* * * * + *
* * * * * +
```

Exemplo: Elementos Iguais em duas listas

- Escrever uma função que recebe 2 listas
- a função deve retornar a lista dos elementos da 2a lista que também estão na primeira (a interseção das duas listas)
- se não tiver nenhum elemento em comum, retornar a lista vazia

Elementos Iguais em duas listas

```
def elemiguais (l1, l2):  
    saida = []  
    for x in l1:  
        if dentro(x,l2):  
            saida.append(x)  
    return saida  
  
def dentro(x, l):  
    for a in l:  
        if a==x:  
            return True  
    return False
```

Exercício

- Escreva um programa que lê 10 números inteiros e os salva em uma lista. Em seguida o programa deve ler um outro número inteiro C . O programa deve então encontrar dois números de posições distintas da lista cuja multiplicação seja C e imprimi-los. Caso não existam tais números, o programa deve informar isto.
- Exemplo: Se $l = [2, 4, 5, -10, 7]$ e $C = 35$ então o programa deve imprimir “5 e 7”. Se $C = -1$ então o programa deve imprimir “Não existem tais números”.