

Universidade do Estado do Amazonas

Escola Superior de Tecnologia

Data: 5 de Fevereiro de 2018

Professora: Elloá B. Guedes

JORNADA PYTHON – SEMANA PEDAGÓGICA IFAM 2018.1

EXEMPLOS DE EXERCÍCIOS

1. (Tipos e Operações) Escreva um algoritmo que dado x segundos fornecidos pelo usuário, transforme-os em horas e minutos, quando possível. Ex.: 3670 segundos = 1h 1min 10seg; 65 segundos = 0h 1min 5seg; 43 segundos = 0h 0min 43seg.
2. (Tipos e Operações) Faça um programa que leia um número real qualquer e apresente na saída: a parte inteira deste número; a parte fracionária e o arredondamento deste número.

```
numero? 10.98
parte inteira: 10
parte fracionaria: 0.98
arredondamento: 11.0
```

3. (Tipos e Operações) Faça um programa que leia uma string e imprima na saída padrão apenas os caracteres definidos pelos índices fornecidos pelo usuário.

```
string? python
inicio da string? 1
fim da string? 4
ytho
```

4. (Desvio Condicional) Escreva um programa que leia um ponto (x, y) do plano cartesiano e diga se o mesmo está dentro da circunferência de raio 3 e centro no ponto $(x_0 = 0, y_0 = 0)$. O ponto está dentro da circunferência se $(x - x_0)^2 + (y - y_0)^2 < \text{raio}^2$, em que (x_0, y_0) é o ponto no centro da circunferência.
5. (Desvio Condicional) A escala de cores RGB (do inglês, Red Green Blue - Vermelho, Verde e Azul) é uma maneira de definir e utilizar cores em um computador. O

usuário informa três parâmetros: vermelho (R), verde (G), e azul (B). Cada um desses três números devem estar no intervalo $[0,255]$. Faça um programa em Python que leia triplas de parâmetros R, G e B e informe a cor correspondente, levando em conta as regras a seguir:

- Quando $R = 0$, $G = 0$ e $B = 0$, a cor correspondente é Preto;
- Quando $R = 255$, $G = 255$ e $B = 255$, a cor correspondente é Branco;
- Quando R é o maior valor de todos, a cor é um Tom de Vermelho;
- Quando G é o maior valor de todos, a cor é um Tom de Verde;
- Quando B é o maior valor de todos, a cor é um Tom de Azul;
- Quando $R = G = B$, a cor é um Tom de Cinza;
- Para os demais casos, exiba a mensagem “Outra cor!”.

Veja a seguir alguns exemplos de execuções:

```
R: 2
G: 0
B: 155
Cor: Tom de Azul
--- FM----
```

```
R: 0
G: 0
B: 0
Cor: Preto
--- FIM----
```

```
R: 300
G: 100
B: 256
Valores Inválidos
--- FIM----
```

```
R: 1
G: 1
B: 1
Cor: Tom de Cinza
```

--- FIM ---

6. (Comandos de Repetição) Escreva um programa para ler um número inteiro k e imprimir os k primeiros números pares positivos e sua soma.
7. (Comandos de Repetição) Um sistema automatizado de pontuação para jogos de tiro ao alvo captura o local em que o projétil do jogador atingiu sob a forma de coordenada de um ponto no plano cartesiano. A mosca do alvo, representada pela origem do plano, encontra-se na posição $(0.0, 0.0)$. Você deve fazer um programa que contabiliza a pontuação de um jogo de tiro ao alvo. A cada disparo, o sistema recebe a posição em que o jogador atirou (um par de valores separados por vírgulas que representam a posição no plano) e imprime a distância em relação à mosca. Para encerrar o jogo, o jogador deve disparar a mais de 200 centímetros de distância da mosca. Ao final, o sistema deve indicar o número total de disparos válidos e a distância média dos tiros dados.
- Não é permitido usar funções prontas como $\min()$, $\max()$ e $\text{cont}()$;
 - A distância entre os pontos (a, b) e (c, d) é dada por $\sqrt{(a - c)^2 + (b - d)^2}$;
 - Veja o exemplo a seguir.

Entrada	Saída
85 45	96.18
35.0 40.0	53.15
-20 15	25.00
215 181	—
	disparos: 3
	distância: 58.11

8. (Comandos de Repetição) Escreva um programa que receba um código gerado para acesso a um formulário online e verifica se ele é verdadeiro ou falso. Um código verificado como verdadeiro é composto por uma sequência de dígitos que se alternam entre pares e ímpares ou entre ímpares e pares. Não são admitidos pares ou ímpares juntos. Assuma que pelo menos 2 dígitos compõem o código. O código 7852 é verificado como verdadeiro: (7) é ímpar; (8) é par; (5) é ímpar e (2) é par. A saída do programa deve informar se o código é verdadeiro ou falso e o número de algarismos do código.

- a) O tamanho do código é arbitrário. O menor código possui tamanho 2;
 - b) Sua saída deve ser um booleano e um inteiro, um em cada linha;
 - c) Você pode efetuar o processamento da entrada como inteiro ou string, como preferir.
 - d) Exemplos:
 - Entrada: 77; Saída: False, 2;
 - Entrada: 25638; Saída: True, 5.
9. (Comandos de Repetição) Faça um programa em Python que receba um número n e imprima o n -ésimo número da série de Tribonacci. A série de Tribonacci é uma modificação da série de Fibonacci, definida da seguinte forma:
- $T_0 = 0$
 - $T_1 = 1$
 - $T_2 = 1$
 - $T_n = T_{n-1} + T_{n-2} + T_{n-3}$
- Ou seja, de acordo com a série de Tribonacci, há três elementos iniciais e os demais números da sequência são gerados pela soma dos três anteriores. Alguns números dessa sequência são: 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, ...
10. (Turtle) Utilizando o módulo `turtle`, desenhe as seguintes figuras:
- a) Triângulo;
 - b) Quadrado;
 - c) Quadrado, com linhas em zig-zag;
 - d) Pentágono;
 - e) Octógono;
 - f) Estrela;
 - g) Espiral.
11. (Funções) Escreva uma função em Python `evalQuadratic(a,b,c,x)` que retorna o valor $a \cdot x^2 + b \cdot x + c$.

12. (Funções) Defina uma função `ehVogal(char)` que retorna `True` se `char` é uma vogal ('a','e','i','o' ou 'u'), e `False` em caso contrário. Assuma que `char` é um único caractere, que pode estar em caixa alta ou caixa baixa, isto é, as entradas 'A' e 'a' são válidas, por exemplo.
13. (Funções recursivas) Escreva uma função recursiva que implemente o equivalente a função `len` para strings. Isto é, sua função deve receber como entrada uma string e retornar o número de caracteres que esta string possui. Considere o caso base como sendo o da string vazia, cujo comprimento é zero.
14. (Funções recursivas) Escreva um programa recursivo em Python que receba um número inteiro e positivo n e que retorne um triângulo de asteriscos com n linhas, em que o número de asteriscos é igual ao número correspondente da linha. Veja o exemplo a seguir considerando $n = 5$.

```
*  
**  
***  
****  
*****
```

15. (Listas) Assuma que as seguintes definições foram feitas:

```
listA = [1, 4, 3, 0]  
listB = ['x', 'z', 't', 'q']
```

Para cada uma das expressões a seguir, indique o tipo e o valor, quando for possível. Indique as situações em que houver erro.

- a) `listA.sort`
- b) `listA.sort()`
- c) `listA`
- d) `listA.insert(0, 100)`
- e) `listA.remove(3)`
- f) `listA.append(7)`

- g) listA
 - h) listA + listB
 - i) listB.sort() listB.pop()
 - j) listB.count('a')
 - k) listB.remove('a')
 - l) listA.extend([4, 1, 6, 3, 4])
 - m) listA.count(4)
 - n) listA.index(1)
 - o) listA.pop(4)
 - p) listA.reverse()
 - q) listA
16. (Listas) Uma empresa aérea pretende mudar o serviço de embarque de seus passageiros em seus vôos e decidiu que os passageiros serão chamados a embarcar de acordo com uma “fila virtual” na qual os passageiros são colocados, de acordo com a ordem de aquisição das passagens. Contudo, a empresa quer priorizar o embarque de pessoas idosas (acima de 60 anos). No momento do embarque, a empresa chama as pessoas idosas antes dos demais. Embora os idosos sejam priorizados, a empresa quer manter os idosos na mesma ordem de entrada na fila.
- Pede-se que você implemente uma função que, dada uma lista das idades dos passageiros (ordenada segundo a fila de embarque), retorne uma outra lista contendo apenas os índices dos passageiros idosos. Veja o assert abaixo para compreender a especificação:
- ```
fila = [25, 33, 67, 61, 35, 8, 12, 15, 22, 63, 75, 30, 34]
assert indices_de_idosos(fila) == [2, 3, 9, 10]
```
17. (Listas) Escreva a função filtraLista(num, lista) que receba um número inteiro e uma lista não vazia de inteiros não-negativos. A função deve retornar uma nova lista apenas com os elementos onde a posição do elemento na lista original é divisível por num. Considere que 0 é divisível por qualquer número e de que num > 0. Veja os asserts a seguir:

```
lista1 = [0,1,2,3,4,5,6]
lista2 = [2,3,5,7,11,13,17]
assert filtraLista(2, lista1) == [0,2,4,6]
assert filtraLista(3, lista1) == [0,3,6]
assert filtraLista(4, lista2) == [2,11]
assert filtraLista(40, lista2) == [2]
```

18. (Listas) Leia uma lista de 10 números inteiros e positivos do usuário. A partir desta lista, obtenha uma lista contendo a soma dos divisores de cada número, incluindo 1 e o próprio número. Exiba os números lidos inicialmente de acordo com a ordem crescente de seus divisores. Veja o exemplo a seguir.

```
antes
lista1 = [3,7,100, 14, 22, 113, 50]
somaDiv = [4, 8, 217, 24, 36, 114, 93]

depois
lista1 = [3, 7, 14, 22, 50, 113, 100]
somaDiv = [4, 8, 24, 36, 93, 114, 217]
```

19. (Dicionários) Assuma que a seguinte definição foi feita:

```
animals = {'a': 'aardvark', 'b': 'baboon', 'c': 'coati'}
animals['d'] = 'donkey'
```

Informe quais são os elementos resultantes das seguintes operações e os seus respectivos tipos. Quando a operação não puder ser realizada, indique erro.

- a) animals
- b) animals['c']
- c) animals['donkey']
- d) len(animals)
- e) animals['a'] = 'anteater'  
animals['a']
- f) len(animals['a'])
- g) animals.has\_key('baboon')
- h) 'donkey' in animals.values()

- i) `animals.has_key('b')`
- j) `animals.keys()`
- k) `del animals['b']`  
`len(animals)`
- l) `animals.values()`

20. (Dicionários) Considere as seguintes expressões:

```
animals = { 'a': ['aardvark'], 'b': ['baboon'], 'c': ['coati']}
animals['d'] = ['donkey']
animals['d'].append('dog')
animals['d'].append('dingo')
```

- a) Escreva uma função denominada `quantos(dicionario)` que recebe um dicionário e retorna a quantidade de valores associada ao dicionário. Para a chamada `quantos(animals)`, o retorno obtido deve ser 6.
  - b) Escreva uma função denominada `maiorDeTodos(dicionário)` que recebe um dicionário e que retorna a chave correspondente à entrada com o maior número de valores associados a ela. Se houver mais de uma resposta, retornar qualquer uma delas é suficiente. Para o exemplo em questão, `maiorDeTodos(animals)` a resposta é `'d'`. Se não houver valores no dicionário, a função deve retornar `None`.
21. (Dicionários) Faça uma função que receba um dicionário e retorne um outro dicionário com as chaves e valores invertidos, o dicionário original não deve ser modificado. Veja o exemplo a seguir.

```
assert inverte({'a':1, 'b':2, 'c':3, 'd':4}) ==
 {1: 'a', 2: 'b', 3: 'c', 4: 'd'}
```