

Exercícios de Fixação – Python (Cap. 7 e 8)

NOME: Camilo Bustamante Moreira

DISCIPLINA: Algoritmos

PROFESSOR: Msc. Lucas Gonçalves Nadalete

Instruções de Entrega:

- **Atividade individual;**
- Entrega a ser realizada até o dia **29/04/2016** às **23:00 hrs.**
- Exercícios do Capítulo 7 (7.1 à 7.10) e Capítulo 8 (8.1 à 8.15) do livro "Introdução à Programação com Python - Edição 2".
- A entrega deve ser feita em um arquivo **.PDF** contendo o nome da disciplina, nome do professor, nome do aluno, e os respectivos enunciados e o código das implementações feitas ou respostas escritas.
- Meio de Entrega: **Exclusivamente via Moodle. Entregas via e-mail serão desconsideradas.**
- Padrão do nome do arquivo: **<nome_aluno> - Exercícios de Strings e Funções.**
 - Obs: Ler atentamente os capítulos 7 e 8 antes de fazerem os exercícios.

CAPÍTULO 7

Exercício 7.1 - Escreva um programa que leia duas strings. Verifique se a segunda ocorre dentro da primeira e imprima a posição de início.

1ª string: AABBEFAATT

2ª string: BE

Resultado: BE encontrado na posição 3 de AABBEFAATT

```
frase1 = input("Informe a frase 1: ").lower()
frase2 = input("Informe a frase 2: ").lower()
if frase2 in frase1:
    print("A frase '%s' foi encontrada na posição %i na frase '%s'" %(frase2,
frase1.find(frase2), frase1))
else:
    print("Frase '%s' não encontrada na frase '%s'" %(frase2, frase1))
```

Exercício 7.2 - Escreva um programa que leia duas strings e gere uma terceira com os caracteres comuns às duas strings lidas.

1ª string: AAABCTBF

2ª string: CBT

Resultado: CBT

A ordem dos caracteres da string gerada não é importante, mas deve conter todas as letras comuns a ambas.

```
frase1 = input("Informe a frase 1: ").lower()
frase2 = input("Informe a frase 2: ").lower()
frase3 = ""
for letra in frase1:
    if letra == " ":
        continue
```

```

if letra in frase2:
    frase3 += letra
frase3 = list(frase3)
frase3 = "".join(frase3)
print(frase3)

```

Exercício 7.3 - Escreva um programa que leia duas strings e gere uma terceira apenas com os caracteres que aparecem em uma delas.

1ª string: CTA

2ª string: ABC

3ª string: BT

A ordem dos caracteres da terceira string não é importante.

```

frase1 = input("Informe a frase 1: ").lower()
frase2 = input("Informe a frase 2: ").lower()
frase3 = ""
for letra in frase1:
    if letra == " ":
        continue
    if letra not in frase2:
        frase3 += letra
for letra in frase2:
    if letra == " ":
        continue
    if letra not in frase1:
        frase3 += letra
frase3 = list(frase3)
frase3 = "".join(frase3)
print(frase3)

```

Exercício 7.4 - Escreva um programa que leia uma string e imprima quantas vezes cada caractere aparece nessa string.

String: TTAAC

Resultado:

T: 2x

A: 2x

C: 1x

```

frase = input("Informe uma frase: ").lower()
letraComputada = ""
for letra in frase:
    if letra not in letraComputada:
        print("A letra '%s' aparece %i vezes na frase '%s'." % (letra, frase.count(letra), frase))
        letraComputada += letra

```

Exercício 7.5 - Escreva um programa que leia duas strings e gere uma terceira, na qual os caracteres da segunda foram retirados da primeira.

1ª string: AATTGGAA

2ª string: TG

3ª string: AAAA

```

frase1 = input("Informe a frase 1: ").lower().strip()
frase2 = input("Informe a frase 2: ").lower().strip()
frase3 = ""

```

```
for letra in frase1:
    if letra not in frase2:
        frase3 += letra
print(frase3)
```

Exercício 7.6 - Escreva um programa que leia três strings. Imprima o resultado da substituição na primeira, dos caracteres da segunda pelos da terceira.

1ª string: AATTCGAA

2ª string: TG

3ª string: AC

Resultado: AAAACCAA

```
frase1 = input("Informe a frase 1: ").lower().strip()
frase2 = input("Informe a frase 2: ").lower().strip()
frase3 = input("Informe a frase 2: ").lower().strip()
for indice in range(len(frase2)):
    frase1 = frase1.replace(frase2[indice], frase3[indice])
print(frase1)
```

Exercício 7.7 - Modifique o programa (listagem 7.45) de forma a escrever a palavra secreta caso o jogador perca.

► Listagem 7.45 – Jogo da forca

```
palavra = input("Digite a palavra secreta:").lower().strip() ❶
for x in range(100):
    print() ❷
    digitadas = []
    acertos = []
    erros = 0
    while True:
        senha=""
        for letra in palavra:
            senha +=letra if letra in acertos else "." ❸
        print(senha)
        if senha == palavra:
            print("Você acertou!")
            break
        tentativa = input("\nDigite uma letra:").lower().strip()
        if tentativa in digitadas:
            print("Você já tentou esta letra!")
            continue ❹
        else:
            digitadas += tentativa
            if tentativa in palavra:
                acertos += tentativa
            else:
                erros += 1
                print("Você errou!")
        print("X==:==\nX : ")
        print("X  0  " if erros >= 1 else "X")
        linha2=""
        if erros == 2:
            linha2 = "  |  "
        elif erros == 3:
            linha2 = " \|  "
        elif erros >= 4:
            linha2 = " \\/  "
        print("X%s" % linha2)
        linha3=""
        if erros == 5:
            linha3+= " /   "
        elif erros>=6:
            linha3+= " / \  "
        print("X%s" % linha3)
        print("X\n=====")
        if erros == 6:
            print("Enforcado!")
            break
```

```
palavra = input("Digite a palavra secreta: ").lower().strip()
for x in range(100):
    print()
```

```

digitadas = []
acertos = []
erros = 0
while True:
    senha = ""
    for letra in palavra:
        senha += letra if letra in acertos else "."
    print(senha)
    if senha == palavra:
        print("Você acertou!")
        break
    tentativa = input("\nDigite uma letra: ").lower().strip()
    if tentativa in digitadas:
        print("Você já tentou esta letra!")
        continue
    else:
        digitadas += tentativa
        if tentativa in palavra:
            acertos += tentativa
        else:
            erros += 1
            print("Você errou!")
    print("X==:==\nX : ")
    print("x o " if erros >= 1 else "X")
    linha2 = ""
    if erros == 2:
        linha2 = " | "
    elif erros == 3:
        linha2 = " \| "
    elif erros >= 4:
        linha2 = " \\/ "
    print("Z%s" %linha2)
    linha3 = ""
    if erros == 5:
        linha3 += " / "
    elif erros >= 6:
        linha3 += " /\ "
    print("X%s" %linha3)
    print("X\n=====")
    if erros == 6:
        print("Enforcado!")
        print("A palavra era '%s'" %palavra)
        break

```

Exercício 7.8 - Modifique o jogo da forca de forma a utilizar uma lista de palavras. No início, pergunte um número e calcule o índice da palavra a utilizar pela fórmula:

$$\text{índice} = (\text{número} * 776) \% \text{len}(\text{lista_de_palavras})$$

```

lista_de_palavras = ["cavalo", "macumba", "bicicleta", "mochila", "orangotango"]
numero = int(input("Informe um número: "))
indice = (numero * 776) % len(lista_de_palavras)
palavra = lista_de_palavras[indice]
for x in range(100):
    print()
digitadas = []

```

```

acertos = []
erros = 0
while True:
    senha = ""
    for letra in palavra:
        senha += letra if letra in acertos else "."
    print(senha)
    if senha == palavra:
        print("Você acertou!")
        break
    tentativa = input("\nDigite uma letra: ").lower().strip()
    if tentativa in digitadas:
        print("Você já tentou esta letra!")
        continue
    else:
        digitadas += tentativa
        if tentativa in palavra:
            acertos += tentativa
        else:
            erros += 1
            print("Você errou!")
            print("X==:==\nX : ")
            print("x o " if erros >= 1 else "X")
            linha2 = ""
            if erros == 2:
                linha2 = " | "
            elif erros == 3:
                linha2 = " \| "
            elif erros >= 4:
                linha2 = " \\/ "
            print("Z%s" %linha2)
            linha3 = ""
            if erros == 5:
                linha3 += " / "
            elif erros >= 6:
                linha3 += " /\ "
            print("X%s" %linha3)
            print("X\n=====")
            if erros == 6:
                print("Enforcado!")
                print("A palavra era '%s'" %palavra)
                break

```

Exercício 7.9 - Modifique o programa para utilizar listas de strings para desenhar o boneco da forca. Você pode utilizar uma lista para cada linha e organizá-las em uma lista de listas. Em vez de controlar quando imprimir cada parte, desenhe nessas listas, substituindo o elemento a desenhar.

Exemplo:

```

>>> linha = list("X-----")
>>> linha
['X', '-', '-', '-', '-', '-', '-']
>>> linha[6] = "|"
>>> linha

```

```
['X', '-', '-', '-', '-', '-', '-', '|']
>>> "".join(linha)
'X-----|'
```

Exercício 7.10 - Escreva um jogo da velha para dois jogadores. O jogo deve perguntar onde você quer jogar e alternar entre os jogadores. A cada jogada, verifique se a posição está livre. Verifique também quando um jogador venceu a partida. Um jogo da velha pode ser visto como uma lista de 3 elementos, onde cada elemento é outra lista, também com três elementos.

Exemplo do jogo:

```
X | O |
---+---+---
| X | X
---+---+---
| | O
```

Onde cada posição pode ser vista como um número. Confira abaixo um exemplo das posições mapeadas para a mesma posição de seu teclado numérico.

```
7 | 8 | 9
---+---+---
4 | 5 | 6
---+---+---
1 | 2 | 3
```

```
tabuleiro = [
    ["1", "2", "3"],
    ["4", "5", "6"],
    ["7", "8", "9"]
]
print("Use os números para escolher a posição em que jogará: ")
jog1 = True
jogadasJog1 = ""
jog2 = False
jogadasJog2 = ""
gameOver = False
vencedor = ""
possibilidades = ["123", "456", "789", "147", "258", "369", "159", "357"]
while True:
    tabuleiroString = ""
    for linha in tabuleiro:
        tabuleiroString += "".join(linha)
    if tabuleiroString.count("X") >= 3 or tabuleiroString.count("O") >= 3:
        jogadasJog1 = list(jogadasJog1)
        jogadasJog1.sort()
        jogadasJog1 = "".join(jogadasJog1)
        jogadasJog2 = list(jogadasJog2)
```

```

jogadasJog2.sort()
jogadasJog2 = "".join(jogadasJog2)
for byte in possibilidades:
    if byte in jogadasJog1:
        vencedor = "jogador 1"
        jog1 = False
        jog2 = False
        gameOver = True
        break
    elif byte in jogadasJog2:
        vencedor = "jogador 2"
        jog1 = False
        jog2 = False
        gameOver = True
        break
if gameOver:
    for x in range(100):
        print()
    print("O vencedor foi o %s \n" %vencedor)
    for linha in tabuleiro:
        print("%s | %s | %s" % (linha[0], linha[1], linha[2]))
        if linha != tabuleiro[2]:
            print("--+---+--")
        break
if jog1:
    for linha in tabuleiro:
        print("%s | %s | %s" % (linha[0], linha[1], linha[2]))
        if linha != tabuleiro[2]:
            print("--+---+--")
    pos = input("Vez do jogador 1 (O). Escolha a posição: ")
    if int(pos) in range(1,10):
        if pos in tabuleiroString:
            for indiceLinha in range(3):
                for indiceColuna in range(3):
                    if tabuleiro[indiceLinha][indiceColuna] == pos:
                        tabuleiro[indiceLinha][indiceColuna] = "O"
                        jogadasJog1 += pos
                        jog1 = False
                        jog2 = True
            else:
                print("Posição já ocupada.")
                continue
        else:
            print("Posição inválida.")
            continue
    elif jog2:
        for linha in tabuleiro:
            print("%s | %s | %s" % (linha[0], linha[1], linha[2]))
            if linha != tabuleiro[2]:
                print("--+---+--")
        pos = input("Vez do jogador 2 (X). Escolha a posição: ")
        if int(pos) in range(1, 10):
            if pos in tabuleiroString:
                for indiceLinha in range(3):
                    for indiceColuna in range(3):
                        if tabuleiro[indiceLinha][indiceColuna] == pos:

```



```

        tabuleiro[indiceLinha][indiceColuna] = "X"
        jogadasJog2 += pos
        jog1 = True
        jog2 = False
    else:
        print("Posição já ocupada.")
        continue
    else:
        print("Posição inválida.")
        continue

```

Capítulo 8

Exercício 8.1 - Escreva uma função que retorne o maior de dois números.

Valores esperados:

máximo(5,6) == 6

máximo(2,1) == 2

máximo(7,7) == 7

```

def maior(n1, n2):
    if n1 > n2:
        return n1
    elif n1 < n2:
        return n2
    else:
        return n1

```

Exercício 8.2 - Escreva uma função que receba dois números e retorne **True** se o primeiro número for múltiplo do segundo.

Valores esperados:

múltiplo(8,4) == **True**

múltiplo(7,3) == **False**

múltiplo(5,5) == **True**

```

def multiplo(n1, n2):
    if n1 % n2 == 0:
        return True
    else:
        return False

```

Exercício 8.3 - Escreva uma função que receba o lado (l) de um quadrado e retorne sua área ($A = \text{lado}^2$).

Valores esperados:

área_quadrado(4) == 16

área_quadrado(9) == 81

```

def area(lado):
    return lado * lado

```

Exercício 8.4 – Escreva uma função que receba a base e a altura de um triângulo e retorne sua área ($A = (base \times altura)/2$).

Valores esperados:

área_triângulo(6,9) == 27

área_triângulo(5,8) == 20

```
def areaTriangulo(altura, base):  
    return (base * altura) / 2
```

Exercício 8.5 – Reescreva a função da listagem 8.5 de forma a utilizar os métodos de pesquisa em lista, vistos no capítulo 7.

► Listagem 8.5 – Pesquisa em uma lista

```
def pesquise(lista, valor):  
    for x,e in enumerate(lista):  
        if e == valor:  
            return x ❶  
    return None ❷  
L=[10, 20, 25, 30]  
print(pesquise(L, 25))  
print(pesquise(L, 27))
```

Exercício 8.6 – Reescreva o programa da listagem 8.8 de forma a utilizar **for** em vez de **while**.

► Listagem 8.8 – Como não escrever uma função

```
def soma(L):  
    total=0  
    x = 0  
    while x<5:  
        total+=L[x]  
        x+=1  
    return total  
L=[1,7,2,9,15]  
print(soma(L)) ❶  
print(soma([7,9,12,3,100,20,4])) ❷
```

```
def soma(L):  
    total = 0  
    x = 0  
    for x in L:  
        total += x  
    return total  
L = [1, 7, 2, 9, 15]  
print(soma(L))  
print(soma([7, 9, 12, 3, 100, 20, 4]))
```

Exercício 8.7 - Defina uma função recursiva que calcule o maior divisor comum (M.D.C.) entre dois números a e b, onde $a > b$.

$$\begin{aligned}
 &ab=0 \\
 &\frac{a}{b} \int \\
 &\int \\
 &b, a-b \int \int a > b \\
 &\text{mdc} \int \\
 &\text{mdc}(a, b) = \int
 \end{aligned}$$

Onde $a-b \int \frac{a}{b} \int$ pode ser escrito em Python como $a \% b$.

```
def mdc(a, b):
    while (a % b != 0):
        a, b = b, (a % b)
    return b
```

Exercício 8.8 - Usando a função mdc definida no exercício anterior, defina uma função para calcular o menor múltiplo comum (M.M.C.) entre dois números.

$$\begin{aligned}
 &\int a \times b \vee \frac{\int}{\text{mdc}(a, b)} \\
 &\int \\
 &\text{mmc}(a, b) = \int
 \end{aligned}$$

Onde $\int a \times b \vee \int$ pode ser escrito em Python como: $\text{abs}(a*b)$.

```
def mdc(a, b):
    while (a % b != 0):
        a, b = b, (a % b)
    return b
def mmc(a, b):
    return abs(a*b) / mdc(a, b)
```

Exercício 8.9 - Rastreie o programa da listagem 8.12 e compare o resultado com o apresentado na listagem 8.13.

► **Listagem 8.12 – Função modificada para facilitar o rastreamento**

```
def fatorial(n):  
    print("Calculando o fatorial de %d" % n)  
    if n==0 or n == 1:  
        print("Fatorial de %d = 1" % n)  
        return 1  
    else:  
        fat = n*fatorial(n-1)  
        print(" fatorial de %d = %d" % (n, fat) )  
    return fat  
fatorial(4)
```

Que produz como resultado a tela da listagem 8.13.

► **Listagem 8.13 – Cálculo do fatorial de 4**

```
Calculando o fatorial de 4  
Calculando o fatorial de 3  
Calculando o fatorial de 2  
Calculando o fatorial de 1  
Fatorial de 1 = 1  
fatorial de 2 = 2  
fatorial de 3 = 6  
fatorial de 4 = 24
```

Os resultados apresentados são idênticos.

Exercício 8.10 - Reescreva a função para cálculo da sequência de Fibonacci, sem utilizar recursão.

```
def fibonacci(x):  
    f = [1]  
    a = 0  
    for indice in range(x - 1):  
        #f, a = f + a, f  
        #print(f)  
        f.append(f[indice] + a)  
        a = f[indice]  
    return f  
print(fibonacci(7))
```

Exercício 8.11 - Escreva uma função para validar uma variável string. Essa função recebe como parâmetro a string, o número mínimo e máximo de caracteres. Retorne verdadeiro se o tamanho da string estiver entre os valores de máximo e mínimo, e falso em caso contrário.

```
def validaString(texto, min, max):
```

```

texto = texto.strip()
if len(texto) in range(min, max + 1):
    return True
else:
    return False

```

Exercício 8.12 - Escreva uma função que receba uma string e uma lista. A função deve comparar a string passada com os elementos da lista, também passada como parâmetro. Retorne verdadeiro se a string for encontrada dentro da lista, e falso em caso contrário.

```

def isStringInLista(texto, lista):
    texto = texto.strip()
    if texto in lista:
        return True
    else:
        return False

```

Exercício 8.13 - Altere o programa da listagem 8.37 de forma que o usuário tenha três chances de acertar o número. O programa termina se o usuário acertar ou errar três vezes.

► Listagem 8.37 – Adivinhando o número

```

import random
n=random.randint(1,10)
x=int(input("Escolha um número entre 1 e 10:"))
if (x==n):
    print("Você acertou!")
else:
    print("Você errou.")

```

```

import random
n = random.randint(1, 10)
tentativas = 3
tentados = []
while tentativas > 0:
    x = int(input("Escolha um numero entre 1 e 10: "))
    if x in tentados:
        print("Você já tentou esse numero.")
        continue
    tentados.append(x)
    if x == n:
        print("Você acertou!")
        break
    else:
        tentativas -= 1
        if tentativas == 0:
            print("Acabaram suas chances. O número era %i." %n)
            break
    print("Você errou. Você ainda tem %i tentativas." %tentativas)

```

Exercício 8.14 - Altere o programa da listagem 7.45 (ver exercício 7.7), o jogo da forca. Escolha a palavra a adivinhar utilizando números aleatórios.

```
import random
lista_de_palavras = ["cavalo", "macumba", "bicicleta", "mochila", "orangotango"]
numero = random.randint(0, len(lista_de_palavras)-1)
palavra = lista_de_palavras[numero]
for x in range(100):
    print()
    digitadas = []
    acertos = []
    erros = 0
    while True:
        senha = ""
        for letra in palavra:
            senha += letra if letra in acertos else "."
        print(senha)
        if senha == palavra:
            print("Você acertou!")
            break
        tentativa = input("\nDigite uma letra: ").lower().strip()
        if tentativa in digitadas:
            print("Você já tentou esta letra!")
            continue
        else:
            digitadas += tentativa
            if tentativa in palavra:
                acertos += tentativa
            else:
                erros += 1
                print("Você errou!")
            print("X==:==\nX : ")
            print("x o " if erros >= 1 else "X")
            linha2 = ""
            if erros == 2:
                linha2 = " | "
            elif erros == 3:
                linha2 = " \| "
            elif erros >= 4:
                linha2 = " \\/ "
            print("Z%s" %linha2)
            linha3 = ""
            if erros == 5:
                linha3 += " / "
            elif erros >= 6:
                linha3 += " /\ "
            print("X%s" %linha3)
            print("X\n=====")
            if erros == 6:
                print("Enforcado!")
                print("A palavra era '%s'" %palavra)
                break
```

Exercício 8.15 - Utilizando a função **type**, escreva uma função recursiva que imprima os elementos de uma lista. Cada elemento deve ser impresso separadamente, um por linha. Considere o caso de listas dentro de listas, como `L = [1, [2, 3, 4, [5, 6, 7]]]`. A cada nível, imprima a lista mais à direita, como fazemos ao indentar blocos em Python. Dica: envie o nível atual como parâmetro e utilize-o para calcular a quantidade de espaços em branco à esquerda de cada elemento.