

Exercícios de Fixação – Python (Cap. 6)

NOME: Camilo Bustamante Moreira

DISCIPLINA: Algoritmos

PROFESSOR: Msc. Lucas Gonçalves Nadalete

Instruções de Entrega:

- **Atividade individual;**
- Entrega a ser realizada até o dia **06/04/2016 às 23:00 hrs.**
- Exercícios do Capítulo 6 (6.2, 6.3, 6.4, 6.7, 6.10 e 6.13) 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 soluções dos exercícios em sequência;
- Meio de Entrega: **Exclusivamente via Moodle.**
- **Obs: Atividades entregues por email serão desconsideradas.**

6.2) Faça um programa que leia duas listas e que gere uma terceira com os elementos das duas primeiras.

```
lista1 = []
lista2 = []
lista3 = []
a = ""
x = 1
while x <= 3:
    a = input("Entre com o %iº elemento da 1ª lista: " %(x))
    lista1.append(a)
    x += 1
print("Lista 1 = %s" %lista1)
x = 1
while x <= 3:
    a = input("Entre com o %iº elemento da 2ª lista: " %(x))
    lista2.append(a)
    x += 1
print("Lista 2 = %s" %lista2)
x = 0
while x < len(lista1):
    lista3.append(lista1[x])
    x += 1
x = 0
while x < len(lista2):
    lista3.append(lista2[x])
    x += 1
print("Lista 3 = %s" %lista3)
```

6.3) Faça um programa que percorra duas listas e gere uma terceira sem elementos repetidos.

```
lista1 = []
lista2 = []
lista3 = []
a = ""
```

```

x = 1
while x <= 3:
    a = input("Entre com o %iº elemento da 1ª lista: " %(x))
    lista1.append(a)
    x +=1
print("Lista 1 = %s" %lista1)
x = 1
while x <= 3:
    a = input("Entre com o %iº elemento da 2ª lista: " %(x))
    lista2.append(a)
    x +=1
print("Lista 2 = %s" %lista2)
lista12 = lista1 + lista2
x = 0
while x < len(lista12):
    a = lista12[x]
    if a not in lista3:
        lista3.append(a)
    x += 1

print("Lista 3 = %s" %lista3)

```

6.4) O que acontece se não verificarmos se a lista está vazia antes de chamarmos o método pop?

Vai retornar o erro "IndexError: pop from empty list", que você está tentando "excluir" um índice que não existe, pois a lista está vazia.

6.7) Faça um programa que leia uma expressão com parênteses. Usando pilhas, verifique se os parênteses foram abertos e fechados na ordem correta.

Você pode adicionar elementos à pilha sempre que encontrar abre parênteses e desempilhá-la a cada fecha parênteses. Ao desempilhar, verifique se o topo da pilha é um abre parênteses. Se a expressão estiver correta, sua pilha estará vazia no final. Exemplo:

(())	OK
()()()()	OK
()	Erro

```

exp = ""
while True:
    exp = input("Entre com uma expressão com parenteses: ")
    if "(" in exp or ")" in exp:
        break
qtdPA = exp.count("(")
qtdPF = exp.count(")")
if qtdPA == qtdPF:
    print("%s OK!" %exp)
else:
    print("%s Erro!" % exp)

```

6.10) Modifique o programa do exercício 6.9* de forma a pesquisar p e v em toda a lista e informando o usuário a posição

onde p e a posição onde v foram encontrados. *O exercício 6.9 refere-se ao exemplo da Listagem 6.23 a seguir:

Listagem 6.23 – Pesquisa sequencial

```
L=[15,7,27,39]
p=int(input("Digite o valor a procurar:"))
achou=False ❶
x=0
while x<len(L):
    if L[x]==p:
        achou=True ❷
        break ❸
    x+=1
if achou: ❹
    print("%d achado na posição %d" % (p,x))
else:
    print("%d não encontrado" % p)
```

```
l = [15, 7, 27, 39]
p = int(input("Digite o valor a procurar: "))
v = int(input("Digite outro valor a procurar: "))
achouP = False
achouV = False
posiP = []
x = 0
while x < len(l):
    if l[x] == p:
        achouP = True
        posiP.append(x)
    x += 1
if achouP:
    print("%i achado na(s) posição(ões) %s" % (p, posiP))
else:
    print("%i não encontrado" % p)
posiV = []
x = 0
while x < len(l):
    if l[x] == v:
        achouV = True
        posiV.append(x)
    x += 1
if achouV:
    print("%i achado na(s) posição(ões) %s" % (v, posiV))
else:
    print("%i não encontrado" % p)
```

6.13) A lista de temperaturas de Mons, na Bélgica, foi armazenada na lista T = [-10, -8, 0, 1, 2, 5, -2, -4]. Faça um programa que imprima a menor e a maior temperatura, assim como a temperatura média.

```
t = [-10, -8, 0, 1, 2, 5, -2, -4]

print(min(t))
print(max(t))
print(sum(t)/len(t))
```

