

Lista 10**Strings****Questão 1**

Qual será o resultado do programa abaixo?

```
1 a = "cavalo"
2 b = "cachorro"
3 c = "gato"
4 d = "cachorro"
5 ##
6 print(type(a))
7 ##
8 print(a is b)
9 print(b is d)
10 print(b is not c)
11 ##
12 print(id(b))
13 print(id(d))
14 ##
15 N = 8
16 M = 8
17 print(N is M)
18 ##
19 print(id(N))
20 print(id(M))
21 print(id(8))
```

Questão 2

Qual será o resultado do programa abaixo?

```
1 singers = "Peter, Paul, and Mary"
2 ##
3 print(singers[0])
4 print(singers[7])
5 print(singers[17])
6 ##
7 print(singers[21])
```

```
8 print(len(singers))
9 ##
10 print(singers[len(singers)-1])
11 print(singers[-1])
12 ##
13 print(singers[0:5])
14 print(singers[7:11])
15 print(singers[17:21])
16 ##
17 singers[0] = 'Z'
18 singers.append("!!")
19 singers += "???"
20 print(singers)
21 ##
22 singers_novo = "R"+singers[1:]
23 print(singers_novo)
24 ##
25 print(singers_novo.replace("?", "!"))
26 print(singers_novo)
```

Questão 3

Qual será o resultado do programa abaixo?

```
1 tipo = "bolo de"
2 fruta = " banana é "
3 valor = 10
4 bolo = tipo+fruta+str(valor)
5 ##
6 print(bolo.upper())
7 print(bolo.lower())
8 print(bolo.capitalize())
9 print(bolo)
10 ##
11 print(bolo*5)
12 print(bolo)
13 ##
14 print(bolo.replace('b', 'B'))
15 print(bolo)
16 ##
17 print(bolo.count('b'))
18 ##
19 bolo.find('b')
20 bolo.find('e')
21 ##
22 p = bolo.find('a')
```

```

23 print(p)
24 p = bolo[p+1:].find('a')
25 print(p)
26 ##
27 novo_bolo = "      "+bolo
28 ##
29 print(novo_bolo)
30 print(novo_bolo.strip())
31 ##

```

Questão 4

Escreva uma função `menor_nome(nomes)` que recebe uma lista de strings com nome de pessoas como parâmetro e devolve o nome mais curto presente na lista.

A função deve ignorar espaços antes e depois do nome e deve devolver o menor nome presente na lista. Este nome deve ser devolvido com a primeira letra maiúscula e seus demais caracteres minúsculos, independente de como tenha sido apresentado na lista passada para a função.

Quando houver mais de um nome com o menor comprimento, a função deve devolver o **primeiro** nome com o menor comprimento presente na lista.

Exemplos:

```

1 menor_nome(['maria', 'josé', 'PAULO', 'Catarina'])
2 >>> 'José'
3 menor_nome(['maria', '  josé  ', '  PAULO', 'Catarina'])
4 >>> 'José'
5 menor_nome(['  Bárbara', '  JOSÉ  ', '  Bill  '])
6 >>> 'José'

```

Questão 5

Escreva uma função em **Python** que recebe um texto T (com espaços) e uma palavra p . Em seguida a função deverá imprimir todas as posições onde ocorrem a palavra p em T e retornar o número de ocorrências de p em T .

Exemplo:

```

1 >>> T = "Duas bananas e 4 abacates. Nao haverá mais bananas."
2 >>> p = "bananas"
3 >>> busca(T, p)
4 5
5 43

```

Questão 6

Qual será o resultado do programa abaixo?

```
1 x = "teoria"
2 y = "pratica"
3 print(x==y)
4 ##
5 print(x is y)
6 ##
7 z = "teoria"
8 print(z == x)
9 ##
10 print(z is x)
11 print(x > y)
12 ##
13 print("a" < "b")
14 print(ord("a"))
15 print(ord("b"))
16 ##
17 print("A" < "b")
18 print(ord("A"))
19 print(ord("b"))
20 ##
21 print("A".lower() < "b".lower())
```

Questão 7

Escreva a função `primeiro_lex(lista)` que recebe uma lista de strings como parâmetro e devolve o primeiro string na ordem lexicográfica. Neste exercício, **desconsidere** letras maiúsculas e minúsculas.

Exemplos:

```
1 primeiro_lex(['olá', 'Azeitona', 'abacate', 'casa'])
2 # deve devolver 'Azeitona'
3 primeiro_lex(['AAAAAA', 'b', 'ABC'])
4 # deve devolver 'AAAAAA'
```

Questão 8

Escreva a função `conta_letras(frase, contar="vogais")`, que recebe como primeiro parâmetro uma string contendo uma frase e como segundo parâmetro uma outra string. Este segundo parâmetro deve ser opcional.

Quando o segundo parâmetro for definido como "vogais", a função deve devolver o numero de vogais presentes na frase. Quando ele for definido como "consoantes", a função deve devolver o número de consoantes presentes na frase. Se este parâmetro não for passado para a função, deve-se assumir o valor "vogais" para o parâmetro.

Exemplos:

```
1 conta_letras('programamos em python')
2 >>> 6
3 conta_letras('programamos em python', 'vogais')
4 >>> 6
5 conta_letras('programamos em python', 'consoantes')
6 >>> 13
```

Questão 9

Escreva a função `maiusculas(frase)` que recebe uma frase (uma string) como parâmetro e devolve uma string com as letras maiúsculas que existem nesta frase, na ordem em que elas aparecem.

Note que para simplificar a solução do exercício, as frases passadas para a sua função não possuirão caracteres que não estejam presentes na tabela ASCII apresentada, como ç, á, Ê, ã, etc.

Exemplos:

```
1 maiusculas('Programamos em Python 3.')
2 >>> 'PP'
3 maiusculas('ProgRamamOS em PYthon!')
4 >>> 'PROPY'
```

Questão 10

Escreva um programa que leia uma string, e verifica se a string lida é ou não um *palindromo*.

Relembrando: Um palindromo é uma palavra ou frase, que é igual quando lida da esquerda para a direita ou da direita para a esquerda (assuma que só são usados caracteres minúsculos e sem acentos. Espaços em brancos devem ser descartados).

```
1 saudavel leva duas
```