

Computação I - Python

Laboratório 5

Seguindo com nossas boas práticas, para cada um dos exercícios a seguir:

- Podem aparecer na especificação do retorno desejado nos enunciados o nome de uma variável ou parâmetro entre os sinais $<$ e $>$, como por exemplo $<var>$. Isso significa que espera-se o valor da variável neste lugar, e não seu identificador.
- antes de começar a escrever código, faça o estudo do problema e o planejamento de sua solução.
- lembre de botar a **documentação** direitinho, dizendo o que a função faz, quais suas entradas e qual o **tipo de dado** de cada entrada, bem como do valor de retorno da função; por exemplo, se sua função recebe dois números inteiros, nos parâmetros chamados a e b e retorna a divisão deles (possivelmente um número fracionário):

```
'''Calcula e retorna a divisão de a por b;  
int, int -> float'''
```

- escolha **nomes elucidativos** para suas funções e parâmetros;
- pense em **valores de teste** relevantes para testar sua função. Ela tem alguma resposta esperada para valores negativos? Valores fracionários? Que tal testar também com valores no extremo do conjunto de dados de interesse da função (maiores valores esperados, menores valores esperados)?
- quando estiver com dificuldade para entender algum erro de funcionamento ou resultado inadequado de sua função, não fique paralizado olhando para a tela! Pegue lápis e papel e recorra ao **teste de mesa**.
- **para fazer a entrega desta atividade prática, escreva suas funções na ferramenta Machine Teaching.**

Vamos lá!

1. Faça uma função que dada uma frase, retorne o número de palavras da frase. Considere que a frase pode ter espaços no início e no final.
2. Dado um texto armazenado em uma string, faça a função que conte o número de frases que aparecem neste texto. Cada frase no texto é terminada com um ponto final, um ponto de exclamação, um ponto de interrogação ou três pontos em sequência (reticências). Pontos de exclamação ou de interrogação não aparecerão repetidos em sequência no texto e esses símbolos só aparecem no texto terminando uma frase. No exemplo a seguir, são contadas 4 frases: “Preciso tirar um cochilo. Meus Deus! Que horas são? Vou perder a minha aula...”
3. Faça uma função que dadas duas listas $L1$ e $L2$ de tamanho 3, gera uma lista $L3$ que é formada intercalando os elementos de $L1$ e $L2$. Exemplo: $L1 = [1, 3, 5]$ e $L2 = [2, 4, 6]$ gera $L3 = [1, 2, 3, 4, 5, 6]$.

4. Faça uma função que, dada uma frase, retorne a frase onde todos os caracteres de pontuação (incluindo travessão, vírgula, dois pontos, ponto e vírgula, além da pontuação de encerramento de frase) tenham sido substituídos por espaço.
5. Faça uma função que dada uma frase retorne uma outra frase que contenha as mesmas palavras da frase de entrada na ordem inversa, sem letras maiúsculas, e sem a pontuação. Dica: remova a pontuação da frase, usando a função que você fez pro exercício anterior. Você pode usar `str.split(string,separador)` para separar as palavras da frase, gerando uma lista, e posteriormente `str.join(separador,lista)` para juntar as palavras de uma lista em uma string. Lembre que a operação de fatiamento pode ser usada pra inverter uma lista, usando -1 como último argumento.

Exemplo:

frase lida: “Nossa, como eu gosto de chocolate.”

frase alterada: “chocolate de gosto eu como nossa”

6. Faça uma função que construa uma pirâmide de números inteiros, dados dois números. Uma pirâmide de números é na verdade uma sequência de números com as seguintes características:
 - o primeiro valor passado como parâmetro deverá ser o primeiro e o último elemento da sequência, marcando o início e fim da sequência;
 - cada valor dentro da sequência não pode ter uma diferença absoluta maior que um de seu vizinho à direita ou à esquerda.
 - o maior número da sequência estará no meio dela, e corresponde ao segundo número dado como entrada.
 - todos os números na pirâmide aparecem duas vezes na sequência com exceção daquele que está no meio.

Sua função receberá como entrada dois números inteiros, e deve retornar uma lista com a sequência correspondente à pirâmide.

Exemplos:

```
>>> piramide(3,6)
[3,4,5,6,5,4,3]
>>> piramide(9,6)
[9,8,7,6,7,8,9]
>>> piramide(6,6)
[6]
```

7. Questão OBI (Olimpíada Brasileira de Informática - OBI2012, Fase 1, Nível 2) -
(http://olimpiada.ic.unicamp.br/pratique/programacao/nivel1/2007f1p1_colisoesColchao)

João está comprando móveis novos para sua casa. Agora é a vez de comprar um colchão novo, de molas, para substituir o colchão velho. As portas de sua casa têm altura H e largura L e existe um colchão que está em promoção com dimensões $A \times B \times C$.

O colchão tem a forma de um paralelepípedo reto retângulo e João só consegue arrastá-lo através de uma porta com uma de suas faces paralelas ao chão, mas consegue virar e rotacionar o colchão antes de passar pela porta.

Entretanto, de nada adianta ele comprar o colchão se ele não passar através das portas de sua casa. Portanto ele quer saber se consegue passar o colchão pelas portas e para isso precisa de sua ajuda.

Entrada: Os parâmetros de entrada são uma lista com as dimensões A , B e C do colchão em centímetros, ordenadas da menor para a maior, e dois inteiros H e L , correspondentes respectivamente a altura e a largura das portas em centímetros.

Saída: A sua função deve retornar **True** se o colchão passa pelas portas e **False** em caso contrário.

Exemplos

Entrada: [25,120,220], 200, 100 ; Saída: **True**

Entrada: [25,205,220], 200, 100 ; Saída: **False**

Entrada: [25,200,220], 200, 100 ; Saída: **True**