



Universidade Federal do Piauí - UFPI
Campus Senador Helvídio Nunes de Barros - Picos-PI
Sistemas de Informação - SI
Programação Orientada a Objeto

Lista 02 – Strings, Listas, Dicionários - 20/11/2020

- Resolva todas as questões em Python.
- Objetivo desta lista é revisar conceitos de programação estruturada e praticar a linguagem Python.
- Data da entrega: 30/11/2020.

1. Aprendemos que a função **len()** retorna o tamanho de uma string. Faça uma função chamada de **tamString(s)** que recebe uma string qualquer e retorna o tamanho de “s” sem usar a função **len()**.
2. Crie uma função que imprima o conteúdo de uma Lista. Por exemplo, se o array for [1, “a”, 2.5, True], então será impresso em cada linha os valores: 1, “a”, 2.5 e True. Construa uma solução que trate a possibilidade que um elemento do vetor também possa ser um vetor, por exemplo: [0,[1,2],3], neste caso a saída do programa em cada linha será: 0,1,2,3. Sua solução está tratando desta possibilidade: [0,1,[2,[3,4]],5] ?
3. Vimos em sala de aula o método **find** e **index** que retornam a posição de um caracter dentro de uma string. Por exemplo: “**Teste**”.**find**(“**T**”), a saída será 0. Implemente sua solução de procurar um caracter em uma string sem utilizar os métodos find ou o index.
4. Faça uma função para retornar uma string ordenada sem usar a função de ordenação disponível. Importante, em Python, dada uma string qualquer, o Python não deixa alterar o conteúdo de uma posição (veja a figura (a) abaixo). Uma alternativa para resolução desta questão é criar uma função que receba uma string e a converta em uma lista. Faça a ordenação usando lista (é possível utilizar os operadores <, > e == para comparar caracteres). Após a ordenação converta a lista em uma String. Abaixo exemplo do erro e como converter string em lista e lista em string.

a. Problema:

```
|>>> a = 'Flavio'
|>>> a[0] = 'H'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

b. Converter uma String em Lista e Converter uma Lista em String:

```
cidade_string = "Picos"
print("Cidade no formato de uma string:", cidade_string)
cidade_lista = list(cidade_string)
print("Uma lista derivada da cidade_string:", cidade_lista)
nova_cidade = ''.join(cidade_lista)
print("Exemplo de como converter uma lista em string:", nova_cidade)
```

```
Cidade no formato de uma string: Picos
Uma lista derivada da cidade_string: ['P', 'i', 'c', 'o', 's']
Exemplo de como converter uma lista em string: Picos
```

5. Faça um joguinho em Python de adivinha. O seu programa vai gerar um número aleatório de 0 até 100, o número é oculto ao jogador. O jogador deverá acertar o número secreto. O jogo terminará se o usuário acertar ou se ele tentar mais de 10 vezes. Quando o usuário acertar o programa deverá perguntar se o jogador deseja repetir ou sair. Toda vez que o usuário informar um número o programa deverá informar qual é o número da tentativa.
6. Escreva um programa que cria uma lista de 10 posições e mostre-a. Em seguida, troque o primeiro elemento pelo o último, o segundo com o penúltimo, o terceiro com o antepenúltimo e, assim, sucessivamente. Mostre a nova lista após todas as trocas.
7. Elabore um algoritmo que leia duas listas de dez posições e faça a multiplicação dos elementos de mesmo índice, colocando o resultado em uma terceira lista, que deve ser mostrada como saída. Não use bibliotecas científicas do Python como numpy. Faça apenas iterando as listas.
8. Gere uma lista de treze elementos inteiros, que é o gabarito de um teste da loteria esportiva, contendo os números 1, 2 ou 3 em cada posição. Também, gere 3 cartões de aposta representando um cartão de um apostador que contem o número do seu cartão e um vetor de respostas com treze posições. Verifique para cada apostador o número de acertos, comparando a lista de gabarito com a lista de respostas. Escreva o número do apostador e o número de acertos. Se o apostador tiver treze acertos, mostre a mensagem "Ganhador".
9. Construa um algoritmo para o funcionamento de uma agenda. Devem ser lidos os seguintes dados de 10 pessoas: nome, endereço, CEP, bairro e telefone. Tais dados devem ser armazenados na agenda, cuja representação é uma lista com 10 linhas (referentes às pessoas) e 5 colunas (referentes aos dados). Por fim, o algoritmo deve gerar como saída a agenda impressa em ordem invertida em relação à ordem de entrada dos dados.

10. Gere uma lista de 100 números inteiros de 1 até 100. Apenas usando as funções básicas (não use bibliotecas científicas tais como numpy, panda ou scipy). Calcule:
- Média
 - Mediana (lembre-se, se o “n” é par ou ímpar)
 - Variância
 - Desvio Padrão
11. Escreva um programa que lê duas notas de vários alunos e armazena tais notas em um dicionário, onde a chave é o nome do aluno. Escreva uma função que retorna a média do aluno, dado seu nome.
12. Uma pista de Kart permite 3 voltas para cada um de 5 corredores. Escreva um programa que leia todos os tempos em segundos e os guarde em um dicionário, onde a chave é o nome do corredor. Ao final diga de quem foi a melhor volta da prova e em que volta; e ainda quem foi o campeão. O campeão é o que tem a menor média de tempos.
13. Escreva um programa para armazenar uma agenda de telefones em um dicionário. Cada pessoa pode ter um ou mais telefones e a chave do dicionário é o nome da pessoa. Seu programa deve ter um menu com as seguintes funções:
- incluirNovoNome – essa função acrescenta um novo nome na agenda, com um ou mais telefones. Ela deve receber como argumentos o nome e os telefones.
 - incluirTelefone – essa função acrescenta um telefone em um nome existente na agenda. Caso o nome não exista na agenda, você deve perguntar se a pessoa deseja incluí-lo. Caso a resposta seja afirmativa, use a função anterior para incluir o novo nome.
 - excluirTelefone – essa função exclui um telefone de uma pessoa que já está na agenda. Se a pessoa tiver apenas um telefone, ela deve ser excluída da agenda.
 - excluirNome – essa função exclui uma pessoa da agenda.
 - consultarTelefone – essa função retorna os telefones de uma pessoa na agenda.
 - Quando o usuário digitar um número negativo o programa é encerrado