Universidade Federal do Rio Grande do Norte Departamento de Computação e Tecnologia DCT1101 - Algoritmos e Lógica de Programação Prof. Amarildo Lucena - Período 2018.1

Quinta Lista de Exercícios

As questões a seguir envolvem o uso das estruturas de lista juntamente com questões lógicas e do dia-a-dia, para os quais será necessário o desenvolvimento de *scripts* em linguagem Python. Em cada caso, procure antes, de fato, desenvolver uma solução para o problema, para só então escrever código, lembrando que, cada problema pode ser resolvido de muitas maneiras diferentes.

1. Escreva um programa em Python que componha uma lista que seja capaz de receber 10 valores, desses, apenas número positivos.
2. Escreva um programa em Python que ordene a lista preenchida na questão anterior. Detalhe, pesquisar sobre o método bolha e como ele funciona.
3. Escreva um programa em Python que simule um jogo de dados, gerando números aleatórios dentro do intervalo 1 - 6, jogue esse "dado" 10 vezes, salve os números que foram sorteados e informe ao usuário qual o maior, o menor e as posições desses números dentro da lista. Detalhe, com e SEM o uso das funções max() e min().
4. Dada a lista [9,1,3,2,5,2,5,5,7,0,4,1,6,5,9] crie um programa em Python que:
   1. Some todos o número da esquerda para a direita.
   2. Subtraia todos os números da direita para a esquerda (Ex: 9-5=4 -> 4-6= -2 ...).
   3. Some a soma dos 3 maiores números com a soma dos três menores.
5. Escreva um programa em Python que, dada uma determinada palavra, verifique se ela é ou não um palíndromo. Uma palavra é dita um palíndromo de ela se mantiver igual quando percorrida pelos dois lados, gerando o mesmo resultado. Ex: ama, arara, aba e etc.
6. Veja o código a seguir e preencha os blocos de código (funcionalidade comentada) e por fim exiba as listas no fim da execução. Obs.: os condicionais dos Ifs podem ser criados a escolha do aluno.

1. candidatos = [22,45,78,5,45,63,99,2,51,8]  
2. aprovados = []  
3. em\_espera = []  
4. if [**bloco de código 1**]:  
5. print("aprovado")  
6. [**bloco de código 2**] #insere na lista de aprovados  
7. elif [**bloco de código 3**]:  
8. print ("lista de espera")

9. [**bloco de código 4**] #insere na lista de espera  
10. else:  
11. print ("quem não estiver nas listas, foi reprovado")  
print("FIM!")

1. Escreva um programa em Python que armazene 10 inteiros positivos diferentes, obtenha a média desses números e exiba todos os números abaixo dessa média.
2. Escreva um programa em Python que mantenha duas listas, uma para contatos e outra para seus respectivos números, insira 5 contados nessa lista e exiba ela no fim da execução, detalhe, o número precisa ter somente 8 dígitos, não mais e nem menos.
3. O que o seguinte código faz?

1. frase = list ("o rato roeu a roupa do rei de roma")

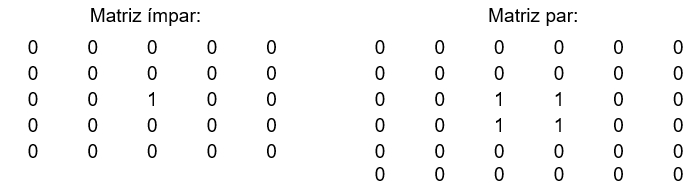
2. print ("frase:", frase)

4. for x in range(len(frase)):

6. if (frase[x] == ' '):

7. frase[x] = '-'

8. print ("frase:", frase)

1. Escreva um programa em Python que preencha uma lista com uma palavra ou qualquer sequência de caracteres e conte para o usuário quantas vogais e quantas consoantes existem ali.
2. Implemente uma lista de listas em Python que, em uma posição guarde nomes femininos em uma lista e nomes masculinos em outra, até que a palavra “sair” seja digitada pelo usuário.
3. Crie uma matriz *n por n* em Python e: 1) preencha a mesma com números aleatórios 2) exiba a soma da diagonal principal com a secundária e 3) o maior e menor elemento da matriz.
4. Crie mais uma matriz quadrada *q*, dessa vez completando-a inicialmente com apenas um valor *n*, verificando se o *q (colunas e linhas)* da matriz QUADRADA é ímpar ou par, faça:
   1. Se ímpar, preencha o valor central da matriz com um elemento diferente do restante.
   2. Se par, preencha os quatro valores centrais da matriz com um elemento diferente do restante.
5. Crie um programa em Python que data uma lista preenchida aleatoriamente, retorne para o usuário os múltiplos de um número *n* passado pelo usuário.
6. Crie um programa em Python que simule um To-Do List, onde cada tarefa a ser feita seja inseria em uma lista no padrão **identificador, descrição, status** (1 para feito e 0 para pendente.), o modelo a ser usado deve ser EXCLUSIVAMENTE uma lista de listas.
7. Questão
8. Questão
9. Questão
10. Questão
11. Questão
12. Questão
13. Questão
14. Questão
15. Questão
16. Questão
17. Questão
18. Questão
19. Questão
20. Questão
21. Questão