

Lista 1 - Exercícios de Python

- 1) O código de César é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes. Por exemplo, com uma troca de três posições, 'A' seria substituído por 'D', 'B' se tornaria 'E', e assim por diante. Implemente um programa que faça uso desse Código de César (3 posições), entre com uma string e retorne a string codificada. Exemplo:

String: a ligeira raposa marrom saltou sobre o cachorro cansado

Nova string: D OLJHLUD UDSRVD PDUURP VDOWRX VREUH R FDFKRUUR FDQVDGR

- 2) Faça um programa em Python para ler uma tabela contendo os nomes dos alunos de uma turma de 5 alunos. O programa deve solicitar ao usuário os nomes do aluno, sempre perguntando se ele deseja inserir mais um nome na lista. Uma vez lidos todos os alunos, o usuário irá indicar um nome que ele deseja verificar se está presente na lista, onde o programa deve procurar pelo nome (ou parte deste nome) e se encontrar deve exibir na tela o nome completo e o índice do vetor onde está guardado este nome.
- 3) Faça um programa em Python que encontre o conjunto de 5 dígitos consecutivos na sequência abaixo que gere o maior produto:

```
73167176531330624919225119674426574742355349194934
96983520312774506326239578318016984801869478851843
85861560789112949495459501737958331952853208805511
12540698747158523863050715693290963295227443043557
66896648950445244523161731856403098711121722383113
62229893423380308135336276614282806444486645238749
30358907296290491560440772390713810515859307960866
70172427121883998797908792274921901699720888093776
65727333001053367881220235421809751254540594752243
52584907711670556013604839586446706324415722155397
53697817977846174064955149290862569321978468622482
83972241375657056057490261407972968652414535100474
82166370484403199890008895243450658541227588666881
16427171479924442928230863465674813919123162824586
17866458359124566529476545682848912883142607690042
24219022671055626321111109370544217506941658960408
07198403850962455444362981230987879927244284909188
84580156166097919133875499200524063689912560717606
05886116467109405077541002256983155200055935729725
71636269561882670428252483600823257530420752963450
```

- 4) Escreva uma função em Python, chamada `contaVoigais`, que conta todas as vogais presentes no texto recebido como parâmetro e retorna um dicionário contendo a quantidade de cada vogal. Escreva um programa a fim de testar sua função, e exiba, no fim, os dados do dicionário retornado.
- 5) Três tipos de dados fundamentais em Python são as listas (“lists”), sequências ou ‘tuplas’ (“tuples”) e dicionários (“dictionaries”). A respeito dessas estruturas, é correto afirmar:
- a) Listas não podem ser modificadas depois de criadas, ao passo que sequências e dicionários podem.
 - b) Listas podem ser modificadas, mas seu tamanho não pode ser modificado após a criação, ao passo que sequências e dicionários não têm essa limitação.
 - c) Dicionários não podem ser modificados depois de criados, ao passo que listas e sequências podem.
 - d) Listas e sequências são indexadas por inteiros, ao passo que dicionários podem ser indexados por strings.
 - e) Listas e dicionários são indexados por inteiros, ao passo que sequências podem ser indexadas por strings.
- 6) Escreva um programa em Python 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 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.
- 7) A conta do cartão de crédito de uma pessoa pode ser modelada por um dicionário com os campos `saldo`, com o saldo devedor da conta, `transações`, com o número de transações que gerou esse saldo, e `média`, com a média de gastos por transação.

Escreva uma função compra em Python, que recebe como parâmetros o dicionário com a conta e o valor da compra e retorna um novo dicionário para aquela conta, com o saldo devedor, número de transações e média de gastos atualizados.

- 8) Faça um programa em Python que leia nomes de alunos e suas respectivas notas até que o nome 'oooo' seja informado, após o fim da leitura, imprima o nome do aluno que possui a maior nota. Obs.: Use dicionário para resolver essa questão.
- 9) Escreva uma função “inva” em Python que recebe um dicionário d e retorna um dicionário “inverso” do dicionário dado, onde, a cada valor v de d está associada a lista das chaves de d que levam a v. Exemplos:

```
>>> inva( {1:2, 3:1, 4:2} )
{2: [1, 4], 1: [3]}
>>> inva( {} )
{}
>>> inva( {2:1, 1:2} )
{1: [2], 2: [1]}
```

- 10) Escreva um programa em Python que:
- lê duas notas de vários alunos e armazena tais notas em um dicionário, onde a chave é o nome do aluno. A entrada de dados deve terminar quando for lida uma string vazia como nome.
 - Escreva uma função que retorna a média do aluno, dado seu nome.
 - Escreva um programa que exiba o nome e a média do aluno de maior média.