

EXERCÍCIO AVALIATIVO – 4 PONTOS

- 1) **Faça um programa** que leia 2 (duas) strings e em seguida informa o conteúdo delas seguido do seu comprimento. Informe também se as duas strings possuem o mesmo comprimento e se são iguais ou diferentes no conteúdo.

Exemplo:

```
Compara duas strings
String 1: Brasil Hexa 2022
String 2: Brasil! Hexa 2022!
Tamanho de "Brasil Hexa 2022": 16 caracteres
Tamanho de "Brasil! Hexa 2022!": 18 caracteres
As duas strings são de tamanhos diferentes.
As duas strings possuem conteúdos diferentes.
```

- 2) **Crie um programa** que solicite o nome do usuário e o imprima no formato de escada invertida.

Exemplo:

```
Digite seu nome: FULANO
FULANO
 ULANO
  LANO
   ANO
    NO
     O
```

- 3) **Desenvolva um programa** que leia 9 (nove) números inteiros e os armazene numa matriz 3x3, em seguida mostre na tela os elementos que estão no triângulo inferior direito da matriz.

← Triângulo Inferior Direito

- 4) **Faça um programa** que solicite a data de nascimento (DD de MM de AAAA) do usuário e imprime a data no formato DD/MM/AAAA.

Exemplo:

```
Data de Nascimento: 29 de Outubro de 1973
Você nasceu em 29/10/1973.
```

- 5) **Desenvolva um jogo** em que o usuário tenha que adivinhar uma palavra que será mostrada com as letras embaralhadas. O **programa** terá uma lista de palavras e escolherá uma aleatoriamente. O jogador terá três tentativas para adivinhar a palavra. Ao final a palavra sorteada deverá ser mostrada na tela, informando se o usuário ganhou ou perdeu o jogo.
- 6) **Crie uma função** que calcula e retorna a média de uma lista de números de ponto flutuante. Agora faça um programa que recebe a temperatura média de cada um dos meses do ano e armazene-as em uma lista. Em seguida, com o auxílio da função criada por você obtenha e mostre todas as temperaturas acima da média anual, e em que meses elas ocorreram (mostrar o nome do mês por extenso: 1 – Janeiro, 2 – Fevereiro, . . . , 12 – Dezembro).
- 7) **Crie um programa** para gerenciar uma lista de tarefas. O programa deve permitir inserir, excluir e marcar tarefas como realizadas. As opções do programa devem ser apresentadas através de um menu (conforme exemplo). Obs.: as ações de inserir, excluir, marcar tarefa realizada, listar tarefas e o menu devem ser feitas **através de funções criadas por você**.

Exemplo:

```
+-----+
|                                     |
|                               LISTA DE TAREFAS                               |
|                                     |
+-----+
| 1 - Inserir tarefa                 |
| 2 - Marcar tarefa como concluída  |
| 3 - Excluir tarefa                |
| 4 - Listar tarefas                |
| 5 - Sair                          |
+-----+
Digite sua opção: 4

Tarefa 01: Estudar Python
Tarefa 02: Pagar conta no banco
* Tarefa 03: Fazer compras no mercado *
Tarefa 04: Pegar o carro na oficina

* - TAREFA REALIZADA
```

- 8) **Desenvolva um programa** que recebe duas listas de caracteres: lista A e lista B. **Implemente funções** para realizar a união, a interseção e a diferença dos elementos dessas duas listas. Depois das entradas o programa deve apresentar os retornos de cada uma das funções criadas na tela.
- 9) **Escreva um programa** que lê duas notas de vários alunos e armazena estas 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 permita retornar a média do aluno, dado o seu nome como argumento.
- 10) Uma pista de Kart permite 3 (três) voltas para cada um dos 6 (seis) 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, **através de funções criadas por você**, de quem foi a melhor volta da prova e em que volta isso ocorreu; e ainda a classificação final em ordem crescente (do campeão para o último colocado). O campeão é o que teve a menor média de tempos.

11) 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 as seguintes opções (**todas realizadas através de funções próprias criadas por você**):

- a. Incluir Novo Nome – permite acrescentar um novo nome na agenda, com um ou mais telefones.
- b. Incluir Telefone – acrescenta um telefone em um nome já existente na agenda. Caso o nome não exista na agenda, você deve perguntar se a pessoa deseja incluí-lo.
- c. Excluir Telefone – exclui um telefone de uma pessoa que já está na agenda. Se a pessoa tiver apenas um telefone, ela deve ser excluída da agenda.
- d. Excluir Nome – exclui uma pessoa da agenda.
- e. Consultar Telefone – retorna os telefones de uma pessoa na agenda.

12) Implemente uma função que, dado um número n maior que 2, devolve uma String com todos os números primos menores ou iguais a n, separados por espaços em branco.

Exemplos:

```
>>> primos(10)
"2, 3, 5, 7"
>>> primos(23)
"23"
>>> primos(36)
"2 3 5 7 11 13 17 19 23 29 31"
```

13) Crie uma função que, dados uma lista de String e um padrão (uma String), retorne todos os elementos da lista com suas letras trocadas (maiúsculas por minúsculas e vice-versa) e na ordem inversa, menos o padrão. Obs.: É proibido usar qualquer função da biblioteca de Python que inverta listas.

Exemplo:

```
>>> inverteRetira(["oi", "pEdRo", "cAMa"], "oi")
["CamA", "PeDrO"]
```

14) Implemente a função splitChars(String) → (Uppercase, Lowercase, Int, Remainder) que recebe um argumento String e retorna uma tupla com 4 elementos. Esses elementos são:

1. Uppercase: as letras maiúsculas da String, ignorando-se repetição.
2. Lowercase: as letras minúsculas da String, ignorando-se repetição.
3. Int: a soma dos dígitos que aparecem na String.
4. Remainder: os caracteres que não forem letras nem dígitos.

Exemplo:

```
>>> splitChars("Be4Ever-sK8!")
("BEK", "vers", 12, "-!")
```

15) Alguns sites de downloads utilizam protetores de links, daqueles que pedem para cadastrar um celular antes de poder fazer o download. Porém, esses protetores na maioria das vezes simplesmente recebem o site de destino como parâmetro da URL e o redirecionam após o cadastro do celular. **Sua tarefa é criar uma função** chamada `extractURL` que recebe uma String contendo um link protegido e retorna o link de destino.

Observações:

- O link de destino pode ou não vir invertido no link protegido e o site do protetor de links não é sempre o mesmo.
- A solução do problema deve utilizar mais de uma função implementada por você.
- Todas as funções auxiliares para a resolução do problema devem ser criadas por você.

Exemplo:

Link de Entrada: <code>http://www.protetordelinks.com/link/?url=http://www.megaupload.com/?d=E34ZP1CO</code> <code>http://www.loadbr.info/link/?url=OC1PZ43E=d?/moc.daolpuagem.www//:ptth</code> Link de Saida: <code>http://www.megaupload.com/?d=E34ZP1CO</code> <code>http://www.megaupload.com/?d=E34ZP1CO</code>
--

16) Defina uma função `inverterOrdem(String) → String`, que recebe uma string contendo uma frase de palavras separadas pelo caractere " " (branco) e retorna a frase com a ordem das palavras invertida. Obs.: Todas as funções auxiliares para a resolução do problema devem ser criadas por você.

Exemplo:

<pre>>>> inverterOrdem("Romeu e Julieta") "Julieta e Romeu"</pre>
--

17) Implemente uma função que, recebendo um número inteiro positivo com mais de 2 (dois) dígitos, devolve uma tupla cujo primeiro elemento é o menor dígito desse número e cujo último é o maior.

Observações:

- Não utilize listas ou strings para a solução do problema.
- Todas as funções auxiliares para a resolução do problema devem ser criadas por você.

Exemplos:

<pre>>>> digitosExtremos(123456789) (1, 9) >>> digitosExtremos(467347563) (4, 3)</pre>
--

18) Crie a função `busca(String, [String]) → [String]` que recebe uma String e uma lista de Strings e retorna todas as Strings da lista que começam com a String passada como parâmetro.

Observações:

- Não utilize funções de Python para strings para a solução do problema.
- Todas as funções auxiliares para a resolução do problema devem ser criadas por você.

Exemplo:

```
>>> busca ("Maracatu", ["Samba", "Maracatu Rural", "Frevo",  
"Maraca", "Maracatu Nação", "Maravilha", "Cavalo Marinho",  
"Maracatus"])  
  
["Maracatu Rural", "Maracatu Nação", "Maracatus"]
```

19) Implemente uma função chamada `eliminarRepetidos([Int]) → [Int]`, que recebe uma lista de inteiros e devolve a mesma lista, mas sem elementos repetidos, mantendo apenas a primeira ocorrência de cada elemento.

Observações:

- Não utilize strings para a solução do problema.
- Todas as funções auxiliares para a resolução do problema devem ser criadas por você.

Exemplo:

```
>>> eliminarRepetidos([1,2,1,2,3,4,5,3,7])  
[1,2,3,4,5,7]
```

20) Desenvolva uma função chamada `montar([Char], String) → Bool` que recebe uma lista de caracteres e uma String e retorna True caso seja possível montar a String passada como parâmetro com o conjunto de caracteres (sem repetir caracteres) ou retorne False caso contrário.

Exemplo:

```
>>> montar(['y','t','e','s','m','t','e','a','k','s'], "testes")  
True  
>>> montar(['a','b','c','l','e','f','h','l','k','s'], "python")  
False
```