FACULDADE DE TECNOLOGIA SENAI/SC FLORIANÓPOLIS

Curso:	Análise e Desenvolvimento de Sistemas	*Semestre:*	1º
Unidade Curricular:	Algoritmos e Programação		
Competência(s):	Codificar programas computacionais utilizando lógica de programação e respeitando boas práticas de programação.		
Docente:	Prof. Luciano Antonio Costa, Dr.		

AULA 06

CONTEÚDO

- Estruturas de dados homogêneas: vetores e matrizes

LISTAS EM PYTHON

DIFERENÇA DA LISTA PARA VETOR E MATRIZ

Em Python, temos como estrutura de dados básica a lista, ao invés de um vetor.

As principais diferenças são:

- O vetor é finito, tem um tamanho definido e não expande, nem reduz. A lista é flexível, posso incluir novos espaços ou remover.
- A ordem do vetor é de suas posições e elas são fixas (pois ele é finito), por exemplo em um vetor de 5 posições podemos inserir um valor na 3ª posição e ele permanecerá na 3ª posição até que façamos alguma ação nele. Já na lista, por conta de sua capacidade de mudança, se inserirmos algo na 3ª posição, este elemento mudará de posição se removermos o elemento a frente, por exemplo o da 1ª posição.
- Um vetor somente aceita elementos do mesmo tipo, ele é homogêneo. Já a lista não apresenta essa restrição.

MANIPULAÇÃO DE UMA LISTA

```
# criação da lista
despesas = []

# inclusão de elementos ao final
despesas.append(20)   # carvão
despesas.append(100)  # picanha
despesas.append(20)   # abobrinha
despesas.append(200)  # milho
print(despesas)
# Saída > [20, 100, 20, 200]

# ler valor de uma posição
print("Custo do carvão:", despesa[0])
# Saída > [20, 100, 20, 200]
```

```
# modificar valores
despesas[3] = 22 # valor do milho estava errado
print(despesas)
# Saída > [20, 100, 20, 22]

# remover valor por posição
del despesas[2] # remover a abobrinha
print(despesas)
# Saída > [20, 100, 200]

# inserir elemento em uma posição
despesas.insert(2, 40) # adicionar a linguiça
print(despesas)
# Saída > [20, 100, 40, 22]

# saber o tamanho da lista
print("Quantidade de elementos: ", len(despesas))
# Saída > Quantidade de elementos:  4

# verificar se algo existe na lista
print(100 in despesas)
# Saída > True

# verificar se algo NÃO existe na lista
print(333 not in despesas)
# Saída > True
```

LAÇO EM UMA LISTA

Como o tamanho da lista é definido, bem como o elemento inicial, podemos fazer uso do seguinte laço:

```
frutas = [ "LARANJA", "BANANA", "MELANCIA", "UVA" ]

# para cada fruta na lista de frutas, faça
for fruta in frutas:
    print("A fruta é:", fruta)
    if fruta in [ "BANANA", "MELANCIA" ]:
        print("Adoro!")
    else:
        print("Quero suco!")

# Saída >
#         A fruta é: LARANJA
#         Quero suco!
#         A fruta é: BANANA
#         Adoro!
#         A fruta é: MELANCIA
#         Adoro!
#         A fruta é: UVA
#         Quero suco!
```

EXERCÍCIOS

INSTRUÇÕES

Para cada exercício a seguir, defina um fluxograma antes de realizar sua implementação.

EXERCÍCIO 01 – NÚMEROS PRIMOS

Implemente o algoritmo do Crivo de Eratosthenes.

EXERCÍCIO 02 – CRIPTOGRAFIA

Implemente uma versão do algoritmo ROT13.

EXERCÍCIO 03 – MATRIZ TRANSPOSTA

Implemente um algoritmo para realizar a transposição de uma matriz bidimensional.

EXERCÍCIO 04 – MODA E MÉDIA

Implemente um algoritmo para cálculo da moda e da média em uma lista.

EXERCÍCIO 05 – MMC E MDC

Implemente um algoritmo para cálculo o MMC e o MDC de uma lista de números.

EXERCÍCIO 06 – BUSCA E CONTAGEM

Dada uma matriz bidimensional que contenha os nomes de frutas, conte a frequência de cada fruta e apresente ao final o totalizador.

Exemplo:

```
[
   [ 'banana', 'uva', 'melancia' ],
   [ 'melancia', 'kiwi', 'laranja' ],
   [ 'kiwi', 'laranja', 'melancia' ]
]
# Saída > banana: 1, uva: 1, melancia: 3...
```