AULA 6

Raciocínio Algorítmico 2024

Estruturas compostas

- Até o momento, nós trabalhamos com variáveis capazes de armazenar um único valor
- A partir de hoje, nós trabalharemos com estruturas compostas, que permite armazenar múltiplos valores em uma única variável
- Isso inclui:
  - Vetores
  - Listas
  - Matrizes

Vetor (Array)

Definição: conjunto ordenado finito de elementos homogêneos.

- **Ordenado**: porque os elementos de um vetor estão dispostos de forma que há um primeiro, um segundo, etc;
- **Finito**: porque todo vetor precisa ter especificado o seu tamanho;
- **Homogêneo**: porque todos os elementos de um vetor possuem o mesmo tipo de dado (int, float, bool, etc)

Declaração

- Exemplo: **v = [1 , 2, 3]**
- Acima, temos um vetor **v** com **3 posições**, e cada uma delas é um número **inteiro**

Acesso

- Para acessar uma posição do vetor, precisamos especificar a posição que desejamos:

v = [1,2,3,4,5]
print(v[0]) -> 1
print(v[1]) -> 2
print(v[0] + v[3]) -> 5
print(v[0+3]) -> 4

- Representação **gráfica**
- **Observação:** as posições começam em **zero**!

Substituir um elemento

- Para substituir uma posição de um vetor, basta atribuir um valor indexando:
- Exemplo:

```
v = [1,2,3,4,5]
print(v) -> [1,2,3,4,5]
v[2] = 1
print(v) -> [1,2,1,4,5]
```

Vetor (cont.)

- Percorrendo cada posição de um vetor:
- Exemplo:

```
frutas = ['laranja', 'maça', 'pera',
'banana', 'kiwi', 'maça', 'banana']
indice = 0
while indice < len(frutas):
    print(frutas[indice])
    indice = indice + 1
```

String

- Uma string é um conjunto de caracteres, no qual podemos indexar para acessar individualmente cada caracter

```
nome = "Vilmar Abreu Junior"
indice = 0
while indice < len(nome):
    print(nome[indice])
    indice += 1
```

Exercícios

1) Preencha um vetor de inteiros com 10 valores lidos pelo teclado, ao fim do programa, imprima cada posição do vetor

2) Implementa um programa que recebe uma String como entrada, conte quantas vogais tem na String.

3) Dado um vetor de inteiros de 10 posições com valores lidos pelo teclado, desenvolver um programa para, em loop:
   Pedir ao usuário um valor N
   Pedir ao usuário um valor de índice
   Atualizar o conteúdo da lista no índice com o valor N, e exibir a nova lista

3) Implemente um programa em Python para verificar quantos números uma aposta acertou na Megasena. O programa deve ler do teclado os 6 números apostados e comparar com 6 números sorteados. Ao final o programa deve exibir os números sorteados, números jogados e quantidade de acertos.

Vetor em Python

- Vetores são extremamente úteis, e vamos trabalhar com eles nas próximas aulas
- Contudo, há um problema: Python não trabalha com vetores
- Na prática, nós vamos simular vetores ao usar **listas**

Listas

- Nos exemplos anteriores, estávamos lidando com **listas**
- Listas possuem **duas** grandes diferenças em relação a vetores em Python:
  - Listas não necessariamente são homogêneas
  - Listas não possuem um tamanho pré-determinado – é possível adicionar ou remover elementos de acordo com a necessidade (limitado a capacidade da memória RAM)

Listas (cont.)

```
lista = []
lista.append(1.0)
print(lista) -> [1.0]
lista.append('banana')
print(lista) -> [1.0,'banana']
lista.remove(1.0)
print(lista) -> ['banana']
del lista[0]
```

Note que a lista é heterogênea!

Esse método remove um valor e não um índice!

del remove por índice

Listas (cont.)

```
frutas = ['laranja', 'maça', 'pera', 'banana', 'kiwi', 'maça', 'banana']
print(frutas.count('maça'))
print(frutas.count('tangerina'))
print(frutas.index('banana'))
print(frutas.index('banana', 4))  # Busca nova ocorrencia de banana a partir da pos 4
frutas.reverse()
print(frutas)
frutas.append('uva')
frutas.insert(1, "morango")
print(frutas)
frutas.sort()
print(frutas)
print(frutas.pop())
print(len(frutas))
```

Exercícios

1. Escreva um algoritmo que leia um vetor com 4 valores e apresente sua média
2. Escreva um algoritmo que crie uma lista de tamanho 5 e imprima seus valores e em seguida a soma dos valores pares e ímpares
3. Ordene um vetor de 100 números inteiros gerados aleatoriamente.

Desafio

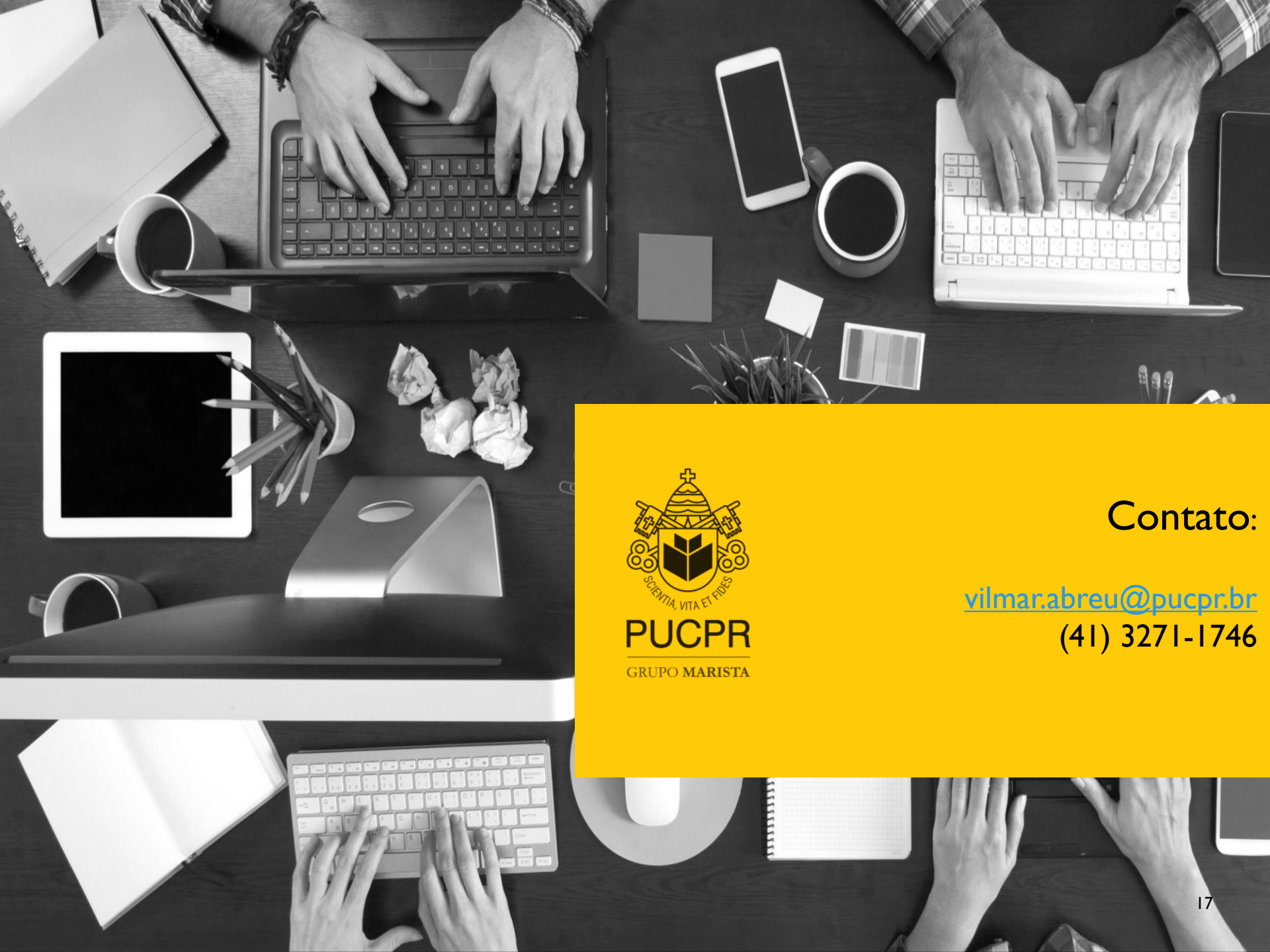
Desenvolva um programa para o "**Jogo da Forca**". Para tal, crie um vetor de strings inicializado com um conjunto de palavras-chave (por exemplo: nomes de capitais do Brasil, ou times de futebol ou Países da América do Sul etc). Sorteie uma das palavras para ser o segredo e forneça seis vidas para o usuário acertar o segredo. A cada rodada informe o número de vidas disponíveis e a disposição das letras acertadas e ausentes na palavra segredo (lembre de quando brincava com este jogo em caderno na infância), mostre também quais as letras que já foram usadas (e não compute acerto ou erro no caso do usuário repetir uma letra já fornecida).

Bibliografia

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos: lógica para desenvolvimento de programação de computadores. 24. ed., rev. São Paulo: Érica, 2010.

MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. 1. ed. São Paulo: Novatec, 2010.



PUCPR

GRUPO MARISTA

Contato:

vilmar.abreu@pucpr.br
(41) 3271-1746