AULA 8

Pensamento Computacional 2023

Campo Minado

O popular jogo Campo Minado (Minesweeper), na opção iniciante, possui uma grade de nove linhas por nove colunas, tendo 10 minas distribuídas aleatóriamente neste campo. Elabore um programa que armazene uma matriz compatível com este cenário, populando-a com minas alocadas randomicamente. Mostre então a distribuição das minas no campo.

Campo Minado (cont.)

Modifique o programa da questão [4] de forma que o usuário forneça a configuração desejada, entre:

1. iniciante: 10 minas em um campo (9x9)
2. intermediário: 40 minas em um campo (16x16)
3. avançado: 99 minas em um campo (16x30)

Depois de popular a matriz com as características compatíveis com a opção do usuário, mostre a distribuição das minas no campo.

Exercício #1

Desenvolva um programa que leia uma matriz quadrada de números inteiros de dimensão (4×4), e então coloque em um outro vetor de 4 posições o maior valor encontrado na coluna da matriz cujo índice é o mesmo do vetor, ou seja, o maior valor da coluna zero da matriz na posição zero do vetor e assim por diante. Mostre então a matriz, o vetor e a média aritmética do vetor.

Exercício #2

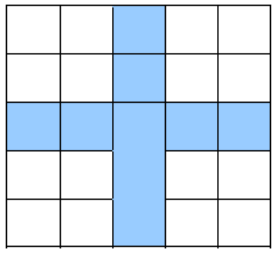
A distância rodoviária entre algumas capitais brasileiras está disponível na tabela abaixo. Para consultar a distância basta cruzar as cidades origem e destino, ou seja, a distância entre Curitiba e São Paulo é de 408 km. Construa um programa que inicialize uma matriz contendo as distâncias apresentadas na tabela acima e que então informe ao usuário a distância necessária para percorrer duas cidades por ele fornecidas.

-x-	Curitiba	Florianópolis	Porto Alegre	São Paulo	Rio de Janeiro
Curitiba	-x-	310	716	408	852
Florianópolis	310	-x-	470	705	1144
Porto Alegre	716	470	-x-	1119	1553
São Paulo	408	705	1119	-x-	429
Rio de Janeiro	852	1144	1553	429	-x-

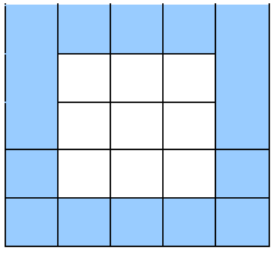
Exercício #3 (Desafio)

Escreva um programa que preencha uma matriz quadrada de números inteiros de dimensão (5×5) com valores inteiros (dentro do intervalo 10 a 99). Para cada uma das figuras abaixo (elabore quatro versões do programa): mostre a matriz original, mostre a matriz apenas com os valores que estão na parte hachurada e mostre a soma destes valores:

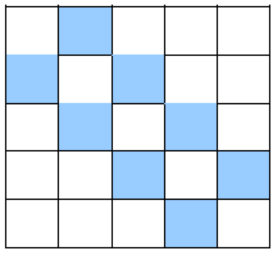
a)



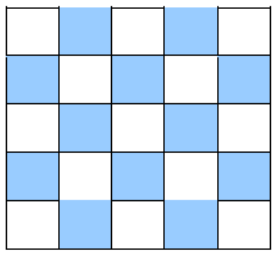
b)



c)



d)



Modularização

- Uma importante característica de bons códigos-fonte é a **modularidade**
- Ser modular significa ser **reutilizável**
- A questão é: **como reutilizar código?**

Funções

- Uma das formas de ter códigos modulares é escrever boas funções
- A definição de uma função é "herdada" da matemática
- Na matemática, uma função é um processo ou relação que mapeia um parâmetro x para um valor y
  - Lembram da notação y = f(x)?
- Na programação: uma função é um processo que a partir de uma ou mais variáveis, retorna um resultado
- y = f(x,z,w,a,...)

Funções (cont.)

- Funções nos permitem “quebrar” um problema complexo em sub-partes mais simples
- Permite reutilizar o código
  - Ex: calcular o troco de 100 faturas diferentes
- Funções deixam nosso código mais legível (impacta em facilidade para identificação de erros, facilidade de manutenção, etc)

Funções nativas do Python

Ao longo da disciplina já utilizamos diversas funções nativas do Python

- print
- input
- round
- randomint

Funções em Python

- Para declarar uma função em Python, usamos o comando **def**
- Exemplo: função de sucessor
  - Entrada (parâmetro): um número qualquer x
  - Saída (resultado): o sucessor de x
- Em Python:

```
def sucessor(x):
    saida = x + 1
    return saida
```

Funções em Python (cont.)

- Notem que o **tipo** de x não é especificado!
  - Se nossa função espera um float/int como entrada e recebe uma string, o que pode acontecer?
  - Maior parte das linguagens requer que o tipo dos parâmetros seja especificado
- A palavra-chave **return** define **o resultado** de uma função
- Todo o conteúdo (bloco) que descreve uma função deve estar **indentado** à direita do comando **def**
- Funções devem ser **declaradas** antes de serem utilizadas

Parâmetros e retorno

- Quantidade de **parâmetros**: 0 a **N**
  Exemplo:
  - ValidarCPF(cpf)
  - Somar(10,20)
  - Multiplicar(10,20,30,40,50)
- Quantidade de **parâmetros**: 0 ou 1
  - Uma função pode **não retornar** nada (sem return)
  - Uma função pode retornar **um valor**
  - Não é uma **boa prática de programação** retornar mais de um valor, apesar do Python permitir

Escopo de variáveis

- Entender o escopo de variáveis é de extrema importância para qualquer programador
- Variáveis podem ser locais ou globais
  - **Locais**:Variáveis definidas dentro de funções
  - **Globais**:Variáveis definidas fora de funções

Escopo de variáveis (cont.)

```
x = 15
if x > 10:
    y = 2
print(x)
print(y)
```

- Flexibilizações do Python, a maioria das linguagens de programação indicaria erro

Escopo de variáveis (cont.)

```
def f(x):
    x = x + 10
    return x

x = 5
y = f(x)
print(x)
print(y)
```

Exercícios

Realize os exercícios abaixo usando funções:

1. Escreva um algoritmo que imprima os 100 primeiros números ímpares.
2. Implemente uma função capaz de calcular o número de segundos de um dia e imprime o resultado na tela.
3. Construa uma função capaz de calcular o consumo de combustível de um veículo a partir da entrada de litros e quilômetros do teclado. A função deve retornar o consumo calculado.

Exercícios

4. Implemente uma função em Python para verificar se um vetor passado como argumento para uma função contém um determinado valor também passado como argumento.

5. Implemente duas funções, uma para inicializar uma matriz recebendo como parâmetro a quantidade de linhas e colunas, e outra função para imprimir a matriz.

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