
Questões práticas – Estruturas Homogêneas (Parte 3 - Matrizes e Vetores)

ATENÇÃO:

- Para esta atividade prática você deverá incorporar, sempre que necessário, as funções apresentadas no *Livro Texto - Módulo 7 - Estruturas Homogêneas*, Seção 7.3: `criarVetor`, `criarMatriz`, `preencherVetor`, `preencherMatriz`, `imprimeVetor`, `imprimeMatriz` e `imprimeMatrizEmLinha`.
- Eventualmente, você deverá criar novas funções, ou modificar as funções apresentadas, para tratar os dados de uma maneira mais específica ou diferente, conforme definições do enunciado da questão.

Questão 1

Na teoria de sistemas, define-se o elemento **MinMax** de uma matriz como sendo o elemento de maior valor da linha onde se encontra o menor valor da matriz. Por exemplo, na matriz a seguir:

$$\begin{bmatrix} 5 & 6 & 3 \\ 8 & 7 & 4 \\ \textit{1} & \mathbf{9} & 2 \end{bmatrix}$$

o menor elemento é o que possui valor **1** (destacado em itálico), que se encontra na linha de índice **2**. Nesta linha, o de maior valor é o **9** (destacado em negrito), que encontra-se na coluna de índice **1**. Implemente um programa que fará a leitura de uma matriz de valores inteiros, usando um `input` e a função `preencherMatriz`. Em seguida, ele chamará a função `linhaMenorValor`, implementada por você, que recebe a matriz e retorna o índice da linha que possui o menor valor de toda a matriz. Considere que todos os valores da matriz são sempre diferentes entre si. Após conhecer o índice da linha, ele chamará a função `colunaMaiorValor`, implementada por você, que recebe a matriz e o índice de uma linha, e retorna o índice da coluna do maior valor constante na referida linha da matriz. Ao final, o programa imprime os índices do **MinMax** e o seu respectivo valor armazenado na matriz. Siga os exemplos de execução.

Exemplo 1:

```
MinMax de uma Matriz
Valores da matriz:
>>> 5,6,3; 8,7,4; 1,9,2
MinMax: [2, 1] = 9
```

Exemplo 2:

```
MinMax de uma Matriz
Valores da matriz:
>>> 10, 14, 2, 4, 6; 5, 1, 11, 20, 18; 21, 22, 23, 24, 25
MinMax: [1, 3] = 20
```

Questão 2

Implemente um programa para apurar as notas dos alunos de BCC701 em uma prova de múltipla escolha. O programa faz a leitura de um vetor contendo o gabarito das questões, ou seja, cada elemento deste vetor contém a resposta correta de uma questão da prova (valores inteiros). Em seguida, o programa faz a leitura das respostas dos alunos em uma matriz. Nesta matriz, as colunas representam as questões e as linhas representam os alunos. Para as duas entradas, além dos `input`'s, são usadas as funções `preencherVetor` e `preencherMatriz`. Se a quantidade de alunos for 0 (string vazia na entrada), o programa imprime a mensagem de erro “Nenhum aluno a avaliar”. Caso contrário, se a quantidade de questões representada na matriz for diferente da quantidade de questões no gabarito, o programa imprime a mensagem de erro “Quantidade de questões incompatível”. Em ambos os casos, o programa imprime a mensagem de erro e é encerrado. Por outro lado, caso as entradas sejam válidas, seu programa chama a função `notasAlunos`, implementada por você, que recebe o vetor de gabarito e a matriz de respostas e preenche um novo vetor contendo as notas dos alunos. Para cada aluno, a nota é obtida nesta função fazendo a comparação entre o gabarito e as suas respostas, contabilizando o número de acertos, a nota final é o número de acertos multiplicado por 10 e dividido pela quantidade de questões. Por fim, o programa principal imprimirá as notas dos alunos usando a função `imprimeVetor`. Siga os exemplos de execução, observe que as notas estão impressas com 2 casas decimais.

Exemplo 1:

```
Notas de BCC701
Digite o gabarito:
>>> B, A, B, E, D
Digite as respostas dos alunos:
>>>
Nenhum aluno para avaliar
```

Exemplo 2:

```
Notas de BCC701
Digite o gabarito:
>>> E, A, A, E
Digite as respostas dos alunos:
>>> E, A, D, E, C, D, B; E, E, E, E, A, D, D; E, E, A, E, A, E, E; E, A, A, E,
    E, E, A; E, E, A, E, E, B, E
Quantidade de questões incompatível
```

Exemplo 3:

```
Notas de BCC701
Digite o gabarito:
>>> D, A, C, E, E
Digite as respostas dos alunos:
>>> D, A, C, C, E; D, A, C, E, A; D, A, C, E, E; E, A, C, E, E; E, E, E, A, A;
    A, B, B, E, C; D, D, C, C, E
Notas dos alunos:
[ 8.00, 8.00, 10.00, 8.00, 0.00, 2.00, 6.00 ]
```

Questão 3

Uma revista sobre automóveis realizará um teste de consumo de combustível e você implementará um programa para ajudá-los. O programa primeiro lê um vetor contendo a capacidade do tanque de combustível em *litros (l)* para cada automóvel testado. Vários condutores participam dos testes, então o programa também lê uma matriz contendo a quantidade de *quilômetros (km)* que cada condutor conseguiu rodar com um tanque de combustível ao conduzir cada automóvel. Nesta matriz, as colunas representam os automóveis e as linhas representam os condutores. Seu programa deve validar as entradas para o vetor e a matriz. Para o vetor, devem ser definidos ao menos um automóvel (a string não pode ser vazia na entrada). Para a matriz, deve haver pelo menos um condutor (a string não pode ser vazia na entrada) e a quantidade de automóveis deve ser igual à quantidade de automóveis representados no vetor. Observe as mensagens de erro nos exemplos de execução. Para preencher o vetor e a matriz, além dos **input's**, são usadas as funções **preencherVetor** e **preencherMatriz**, respectivamente. Se as entradas forem válidas, seu programa chamará a função **calculaConsumos**, implementada por você, que recebe o vetor e a matriz, e preenche uma nova matriz, contendo a divisão entre a quilometragem rodada e a capacidade do tanque para cada condutor, em cada automóvel. O resultado será uma matriz contendo todos os consumos em *km/l*. Este resultado é impresso no terminal usando a função **imprimeMatriz**. Siga os exemplos de execução e observe que nas entradas os valores são sempre inteiros.

Exemplo 1:

```
Teste de consumo
Capacidade dos tanques:
>>>
É necessário pelo menos um automóvel
```

Exemplo 2:

```
Teste de consumo
Capacidade dos tanques:
>>> 50, 60, 45, 80
Quilometragens dos condutores:
>>>
Deve haver pelo menos um condutor
```

Exemplo 3:

```
Teste de consumo
Capacidade dos tanques:
>>> 50, 60, 45, 80
Quilometragens dos condutores:
>>> 100, 200, 300; 150, 250, 350
Quantidade de automóveis incompatível
```

Exemplo 4:

```
Teste de consumo
Capacidade dos tanques:
>>> 50, 60, 75, 80
Quilometragens dos condutores:
>>> 550,660,825,880;530,610,785,845;545,625,805,865;520,630,795,850
Consumos km/l:
[ 11.00, 11.00, 11.00, 11.00 ]
[ 10.60, 10.17, 10.47, 10.56 ]
[ 10.90, 10.42, 10.73, 10.81 ]
[ 10.40, 10.50, 10.60, 10.62 ]
```

Questão 4

Agora você vai aprimorar o programa da questão anterior. Após imprimir a matriz contendo os consumos, seu programa principal chamará a função `calculaMedias`, implementada por você, que recebe a matriz de consumos e preenche um novo vetor, contendo a média de consumo obtido para cada automóvel. O vetor de médias é impresso no terminal usando a função `imprimeVetor`. Siga os exemplos de execução.

Exemplo 1:

```
Teste de consumo
Capacidade dos tanques:
>>> 50, 60, 75, 80
Quilometragens dos condutores:
>>> 550,660,825,880;530,610,785,845;545,625,805,865;520,630,795,850
Consumos km/l:
[ 11.00, 11.00, 11.00, 11.00 ]
[ 10.60, 10.17, 10.47, 10.56 ]
[ 10.90, 10.42, 10.73, 10.81 ]
[ 10.40, 10.50, 10.60, 10.62 ]
Médias dos consumos por automóvel:
[ 10.72, 10.52, 10.70, 10.75 ]
```

Exemplo 2:

```
Teste de consumo
Capacidade dos tanques:
>>> 40, 60, 80
Quilometragens dos condutores:
>>> 400, 600, 800;400, 600, 800;400, 600, 800;400, 600, 800;400, 600, 800
Consumos km/l:
[ 10.00, 10.00, 10.00 ]
[ 10.00, 10.00, 10.00 ]
[ 10.00, 10.00, 10.00 ]
[ 10.00, 10.00, 10.00 ]
[ 10.00, 10.00, 10.00 ]
Médias dos consumos por automóvel:
[ 10.00, 10.00, 10.00 ]
```