

Aula prática - Semana 7 - ERE

Nesta lista, a questão 1 constitui a atividade avaliativa que será utilizada para contabilizar notas de atividades práticas. As demais questões são desafios que não contabilizam nota, mas cuja realização é desejável para exercitar lógica. A questão 1 possui exemplos de execução associados. Note que são apenas exemplos, uma vez que vocês podem propor suas próprias formas de entrada e saída de dados. No entanto, é desejado que se assemelhe ao padrão apresentado no exemplo.

Os seguintes pontos serão considerados na avaliação:

- Atender os requisitos do enunciado.
- Indentação e organização de código.
- Utilização dos tipos corretos de variáveis.
- Utilização correta de constantes nomeadas quando necessário.
- Observação das boas práticas de programação vistas em aula, considerando concisão e eficiência.
- Realização de validação quando a questão requisitar.

1. O mercado das semanas anteriores cresceu e agora se tornou um hipermercado (vendendo inclusive itens de lojas de departamento), possuindo F filiais e vendendo P produtos. Você deve elaborar um programa que auxilie o mercado a analisar as vendas das filiais do mercado ao final de um mês. Ou seja, o programa só deve lidar com a quantidade vendida por cada loja de cada produto em um dado mês. O programa deve:

- Pedir para o usuário informar o nome (máximo de 50 caracteres) de cada um dos P produtos da loja.
- Pedir para o usuário informar o preço de cada um dos P produtos. Assuma que o preço é o mesmo em todas as filiais.
- Pedir para o usuário informar o código de cada um dos P produtos. Assuma que o código de um produto é o mesmo em todas as filiais.
- Pedir para o usuário informar a quantidade de cada produto que foi vendida por cada filial no mês.
- Pedir para o usuário informar o nome de um produto e exibir o código e o preço deste produto. Caso o nome informado seja inválido, o programa deve realizar a consistência, só prosseguindo quando o usuário informar um nome válido.
- Pedir para o usuário informar um código de produto e exibir na tela o nome do produto e o total de unidades vendidas deste produto considerando as vendas de todas as filiais. Se o usuário informar um código inválido, o programa deve realizar a consistência, só prosseguindo quando o usuário informar um código válido.
- Determinar e exibir a média de unidades vendidas de cada produto por filial, exibindo o nome do produto, seu código, seu preço e a média vendida.
- Determinar e exibir o faturamento total de cada filial no mês, considerando o total de produtos vendidos de cada tipo e seu preço.
- Determinar o produto que representou o maior faturamento (considerando total de unidades vendidas em todas as lojas e os preços dos produtos) e exibir seu nome e

código e o total faturado com vendas deste produto. Caso mais de um produto ficar empatado, seu programa deve exibir todos que atingiram esse faturamento máximo.

Considere, por conveniência, P como 3 e F como 3 também, embora o programa deva funcionar normalmente (e de forma consistente) caso o valor dessas constantes seja alterado.

Exemplo de execução:

Suponha que os seguintes dados estão armazenados em memória. Note que esta tabela é só uma representação visual dos dados, no programa, os dados deverão ser representados de forma adequada em memória.

Nome	Código	Preço	Qtd filial 1	Qtd filial 2	Qtd filial 3
Cadeira de escritório	123	200,00	20	30	50
Mesa de escritório	741	1000,00	30	10	5
Sofá	951	3000,00	20	10	10

Exemplo de execução:

Informe o nome de um produto: Geladeira

Nome de produto inválido!

Informe o nome de um produto: Sofá

Código: 951 Preço: 3000.00

Informe o código do produto: 987

Código de produto inválido!

Informe o código do produto: 951

Sofá: Foram vendidas 40 unidades no total

Médias de unidades vendidas de cada produto por filial:

-Cadeira de escritório Código: 123 Preço: 200.00 Média: 33.33

-Mesa de escritório Código: 741 Preço: 1000.00 Média: 15

-Sofá Código: 951 Preço: 3000.00 Média: 13.33

Faturamento total de cada filial:

-Filial 1: R\$ 94000.00

-Filial 2: R\$ 46000.00

-Filial 3: R\$ 45000.00

Produto que gerou o maior faturamento:

Sofá Código: 951 Faturamento: R\$ 120000.00

2. (Desafio 1) Considere N cidades (utilize #define N 5). Faça um programa que utilize uma matriz para representar as distâncias entre essas cidades, de tal modo que a célula (i,j) da matriz representa a distância entre a cidade i e a cidade j. Nesta matriz, se uma célula contém o valor 0 (zero), isso indica que é uma rota para a própria cidade, se ela contém o valor -1, indica que não há via direta que conecta duas cidades, qualquer outro valor positivo indica que há uma via e este valor indica a distância entre as duas cidades. A seguir, o programa deve ler uma sequência de valores (cada valor entre 0 e N-1), onde cada valor indica uma cidade, e a sequência indica um trajeto (ordem de visitação) que alguém gostaria de realizar para visitar um conjunto dessas cidades. O programa deve parar de ler valores quando for informado um valor negativo, ou quando for constatado que o trajeto é impossível, visto que o usuário informou uma cidade que não pode ser acessada a partir da última. O programa deve informar a distância do trajeto válido (até o usuário informar -1). Quando o usuário informar uma cidade que não pode ser acessada a partir da última, ele deve informar isso.

Exemplo de execução:

Matriz de distâncias

0	100	-1	222	321
100	0	50	-1	50
-1	50	0	148	-1
222	-1	148	0	57
321	50	-1	57	0

Exemplo 1:

Informe uma cidade: 0

Informe uma cidade: 3

Informe uma cidade: 2

Informe uma cidade: 1

Informe uma cidade: 4

Informe uma cidade: 3

Informe uma cidade: -1

A distância total do trajeto é de 527

Exemplo 2:

Informe uma cidade: 3

Informe uma cidade: 4

Informe uma cidade: 1

Informe uma cidade: 3

Não há rota entre as cidades 1 e 3.

3. (Desafio 2) Você deve elaborar um jogo parecido com o jogo clássico “campo minado”. O seu campo tem dimensões N X M (definir como constantes). O programa

deve inicialmente pedir para o usuário informar um valor m que define quantas minas ele deve distribuir no campo. A seguir, o programa deve distribuir aleatoriamente pelo campo essas m minas. Você pode assumir que seu campo é uma matriz de inteiros de $N \times M$ (inicialmente inicializada com zero) e definir como -1 o valor das células que vão receber minas. A seguir, o programa deve calcular para cada célula da matriz que representa o campo, quantas minas existem na vizinhança e atribuir esse número como valor da célula. Em uma outra matriz de $N \times M$ (visitados), que inicialmente é inicializada em 0, você deve representar quais células o usuário já visitou. Quando o usuário visita uma célula, esta célula recebe 1. A seguir, o programa deve continuamente pedir para o usuário qual célula ele deseja visitar (linha e coluna da matriz). Caso o usuário indique uma célula com mina, o jogo termina e o usuário perde. Caso o usuário visite uma posição sem mina, deve-se mostrar para o usuário a quantidade de minas na vizinhança daquela célula. Se o usuário visitar todas as posições sem mina, o jogo termina e ele ganha. Quando o jogo termina (com o jogador ganhando ou perdendo), exibir todas as posições que possuem minas. O programa deve informar quando o usuário informar uma célula já visitada também. Alternativamente, você pode exibir uma representação do estado atual do campo, cada vez que pedir para o usuário informar a célula a ser visitada. Por exemplo, uma representação poderia exibir o caractere "#" para células ainda não visitadas e, no caso de uma célula já visitada, exibir o dígito que representa quantas minas existem na vizinhança. Então, em um campo de 5×5 , em um dado momento poderia ser exibido o seguinte estado para o usuário:

```
#####
## 1 ##
## 0 ##
## 2 ##
#####
```

Essa representação indicaria que boa parte do campo não foi visitado ainda, e que algumas posições possuem minas na vizinhança.