

Aula prática - Semana 5 - ERE

Nesta lista, as questões 1 e 2 constituem o conjunto de atividades avaliativas que serão utilizadas para contabilizar notas de atividades práticas. As demais questões são desafios que não contabilizam nota, mas cuja realização é desejável para exercitar lógica. As questões 1 e 2 possuem exemplos de execução associados. Note que são apenas exemplos, uma vez que vocês podem propor suas próprias formas de entrada e saída de dados. O importante o programa realizar o que é pedido.

Os seguintes pontos serão considerados na avaliação:

- Lógica adequada para resolução do problema
- Uso de variáveis adequadas com tipos adequados
- Uso de identificadores significativos para variáveis
- Atender os requisitos do enunciado.
- Indentação e organização de código.
- Uso adequado dos comandos e estruturas vistos em aula
- Utilização correta de constantes nomeadas quando necessário.
- Observação das boas práticas de programação vistas em aula (considerando inclusive concisão e eficiência).
- Realização de validação quando a questão requisitar.

1. Você deve desenvolver um programa para processar vendas em um mercado. O mercado possui os seguintes itens, com os respectivos valores (valores reajustados em função da inflação no período):

Código do produto	Produto	Valor unitário
10	Pão	R\$ 2,25
23	Leite (1 L)	R\$ 7,65
8	Biscoito (pct 300g)	R\$ 6,93
90	Arroz (pct 1kg)	R\$ 30,21
76	Feijão (pct 1kg)	R\$ 12,33
0	Finaliza compra	

O mercado oferece 2% de desconto para compras acima de R\$ 100,00.

O programa deve continuamente processar compras de múltiplos clientes. Para isso, o programa deve exibir um menu de opções (0-Sair 1-Nova compra). Faça consistência para garantir que o programa só prosseguirá caso o usuário informe uma dessas opções. Caso o usuário selecione a opção 0, o programa deve parar de processar novas compras. Caso o programa selecione 1, o programa deve processar uma nova compra. Neste caso, o programa deve exibir a tabela acima para este cliente específico e continuamente pedir para o usuário informar o código do produto (deve ser existente) e a respectiva quantidade desejada (deve ser maior que zero). Caso o código ou a quantidade sejam inválidas, o programa deve informar isso ao usuário e repetir a operação de requisitar essas informações. O cliente indica que finalizou sua compra ao informar o código zero para o produto. Ao fim da compra de cada cliente o programa deve informar o total da compra e, caso ele tenha desconto, informar o total com desconto também (isso não deve ser informado se ele não cumprir o requisito para o desconto). No fim do dia (quando o usuário selecionar a opção Sair no primeiro menu), o programa deve informar as seguintes estatísticas:

- Quantidade de compras realizadas.
- Total faturado no dia (considere os valores pagos incluindo os descontos)
- Média do valor das compras.

Exemplo de execução:

Informe a opção (0-Sair 1-Nova compra): 1

Informe o código do produto: 10
Informe a quantidade do produto 10: 5
Informe o código do produto: 8
Informe a quantidade do produto 8: 4
Informe o código do produto: 3999
Código inválido!
Informe o código do produto: 0
Compra finalizada!
Total: R\$ 38.97

Informe a opção (0-Sair 1-Nova compra): 1

Informe o código do produto: 23
Informe a quantidade do produto 23: 10
Informe o código do produto: 10
Informe a quantidade do produto 10: 0
Quantidade inválida!
Informe a quantidade do produto 10: 20
Informe o código do produto: 90
Informe a quantidade do produto 90: 5
Informe o código do produto: 0
Compra finalizada!
Total: R\$ 272.55
Total com desconto: R\$ 267.10

Informe a opção (0-Sair 1-Nova compra): 3

Opção inválida!

Informe a opção (0-Sair 1-Nova compra): 0

Caixa fechado!

Estatísticas do dia:

- Quantidade de compras realizadas: 2
- Total faturado no dia: R\$ 306.07
- Média do valor das compras: R\$ 153.04

2. Você deve elaborar um programa que simula o deslocamento de um objeto (representado por um par de coordenadas) em um espaço retangular. O programa deve inicialmente ler 4 valores reais que representam respectivamente as coordenadas cartesianas dos pontos $P1(x1,y1)$ e $P2(x2,y2)$. Onde $P1$ representa o ponto do canto inferior esquerdo de um retângulo e $P2$ representa o ponto superior direito deste retângulo. Logo, as coordenadas x e y de $P2$ devem ser maiores que as coordenadas x e y de $P1$, respectivamente. A seguir, o programa deve ler dois valores que representam as coordenadas de um ponto $P3(x3,y3)$, que representa a posição inicial deste objeto. A seguir, o programa deve ler um valor positivo que representa o passo de deslocamento. A seguir, o programa deve continuamente pedir para o usuário informar uma direção de deslocamento e efetuar o deslocamento do objeto, alterando e informando a sua nova posição. O programa deve finalizar quando o objeto sair do perímetro definido pelo retângulo, exibindo uma mensagem que indica isso. Assuma que para sair do perímetro, o objeto deve ter uma posição fora dos limites. Se estiver exatamente sobre o limite ele é considerado dentro do perímetro. As direções são representadas pelas seguintes teclas:

- A(a): anda para a esquerda, conforme o passo definido.
- D(d): anda para a direita, conforme o passo definido.
- S(s): anda para baixo, conforme o passo definido.
- W(w): anda para a cima, conforme o passo definido.

Caso o usuário informe uma direção inválida o programa deve informar isso e continuar a execução exibindo a posição atual e executando a próxima iteração.

Caso o usuário informe algum dos valores iniciais ($P1, P2, P3$ e passo, inválidos, o programa deve ser encerrado, com uma mensagem informando que aquela informação é inválida). Não é necessário deixar o programa repetindo para coletar as informações válidas.

Ao fim do programa, ele deve informar quantos movimentos o objeto realizou para sair do perímetro.

Dica: Você pode fixar as informações que deveriam ser lidas no programa durante o desenvolvimento para facilitar o desenvolvimento da parte mais complicada do código e só incluir as leituras das variáveis no fim do desenvolvimento.

Exemplo de execução:

Informe as coordenadas do ponto $P1$: 0 0

Informe as coordenadas do ponto $P2$: 10 10

Informe as coordenadas do objeto: 5 5

Informe o passo de deslocamento: 1.5

Informe a direção: a

Nova posição: 3.5 5

Informe a direção: w

Nova posição: 3.5 6.5

Informe a direção: k

Direção inválida

Informe a direção: w

Nova posição: 3.5 8

Informe a direção: d

Nova posição: 5 8

Informe a direção: w

Nova posição: 5 9.5

Informe a direção: d

Nova posição: 6.5 9.5

Informe a direção: w

Nova posição: 6.5 11

O objeto saiu do perímetro definido realizando 7 movimentos.

3. (Desafio 1) A **persistência multiplicativa** de um número inteiro é o número de vezes que os dígitos deste número devem ser multiplicados, para que o resultado chegue a um único dígito (um número entre 0 e 9). Você deve fazer um programa que lê um valor inteiro n. Depois disso, o programa deve informar qual é a persistência multiplicativa do número n informado e qual foi o dígito resultante. Note que não sabemos o número de dígitos do número n informado.

Por exemplo, se o usuário informar o número 6423, a persistência multiplicativa é 3 e o dígito resultante é 6, porque:

$$6*4*2*3=144$$

$$1*4*4=16$$

$$1*6=6$$

4. (Desafio 2) Dizemos que um número inteiro n2 está **contido literalmente** em outro número inteiro n1, quando a sequência de dígitos que forma o número n2 é uma subsequência da sequência de dígitos que forma o número n1. Por exemplo: 123 está contido literalmente em 1234, 234 está contido literalmente em 1234, 23 está contido literalmente em 1234, 456 está contido literalmente em 321456987, etc. Faça um programa que lê os números inteiros n1 e n2, tal que n1 deve ser maior ou igual a n2 e que informe se n2 está contido literalmente em n1 ou não. Note que não sabemos qual é o limite máximo de dígitos que n1 e n2 podem possuir.

5. A conjectura de Collatz diz que se escolhermos qualquer número X, se ele for par, dividimos X por 2; se X for ímpar multiplicamos X por 3 e adicionamos 1, realizando o processo repetidamente, em algum momento obtemos o número 1. Note que se trata de uma conjectura, e que portanto ainda não foi provada, apesar dos esforços dos matemáticos. Faça um programa que lê um número x e um número r que indica o máximo de repetições que devem ser feitas. O programa deve informar quantas repetições foram realizadas para chegar a 1, realizando o processo acima, caso o número de repetições seja menor que r. Caso contrário, o programa deve informar que não foi possível chegar a 1 com r repetições. Para os curiosos, [aqui](#) e [aqui](#) tem mais duas referências interessantes sobre a conjectura. Abaixo a equação de recorrência que é utilizada para calcular cada um dos termos:

$$X_n = \begin{cases} \frac{X_{n-1}}{2}, & \text{se } X \text{ for par} \\ 3 \times X_{n-1} + 1, & \text{se } X \text{ for ímpar} \end{cases}$$