

Atividade avaliativa - Semana 4 - Estrutura de repetição FOR

Nesta lista, as questões 1 e 2 constituem o conjunto de atividades avaliativas que serão utilizadas para contabilizar notas de atividades práticas. As questões 3,4,5 e 6 são desafios que não contabilizam nota, mas cuja realização é desejável. As questões 1 e 2 possuem exemplos de execução associados. Note que, a menos que seja pedido, não é necessário que as entradas e saídas sejam exatamente como as dos exemplos.

Critérios de avaliação:

- Uso de variáveis adequadas com tipos adequados
- Uso de identificadores significativos para variáveis
- Uso adequado dos comandos e estruturas vistos em aula
- Programa atende aos requisitos do enunciado.
- Lógica adequada para resolução do problema
- Busque desenvolver o algoritmo/programa mais eficiente possível
- Busque desenvolver o algoritmo/programa mais conciso possível (evitando redundância de código).

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

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

O programa deve inicialmente exibir para o usuário a tabela de itens que o mercado possui. Procure utilizar uma formatação semelhante à do exemplo. A seguir, o programa deve requisitar que o usuário informe quantos tipos de itens ele deseja comprar. A seguir, o programa deve requisitar que o usuário informe o código e a quantidade de cada tipo de item que ele deseja. Ao fim, o programa deve informar para o usuário o valor que ele deve pagar. Informe o valor original e o valor com desconto, caso seja aplicável (caso não tenha desconto, exiba apenas o total). Caso o usuário informe que, por exemplo, deseja comprar 3 tipos de itens diferentes, mas em um dos itens ele informa um código inválido, o programa só deve processar os dois itens para os quais ele informou códigos válidos. O desenvolvedor sabe que isso é meio estranho, mas ele ainda está aprendendo a programar. Ele vai deixar o programa melhor na próxima semana. O programa também não deve controlar se o usuário informou o mesmo código mais de uma vez. Isso é um problema do usuário.

Exemplo de execução:

Informe a quantidade de itens que você deseja comprar: 4

Informe o código do item 1: 76

Informe a quantidade do item 1: 5

Informe o código do item 2: 223

Código inválido!

Informe o código do item 3: 10

Informe a quantidade do item 3: 10

Informe o código do item 4: 90

Informe a quantidade do item 4: 5

Total: R\$ 235.20

Total com desconto: R\$ 230.50

2. Você deve escrever um programa para suportar a realização de uma pesquisa. O programa deve inicialmente ler a quantidade de pessoas que serão entrevistadas. A seguir, para cada pessoa o programa deve ler a idade, a renda mensal e a localidade (U - urbana, ou R - rural). Não é necessário realizar consistência de dados de entrada. Ao fim, o programa deve informar:

- Qual a maior e a menor renda mensal dentre as informadas.
- Percentual de entrevistados de áreas urbanas e de áreas rurais que foram entrevistadas.
- A média da renda mensal recebida por pessoas de áreas urbanas e por pessoas de áreas rurais.

Exemplo de execução:

Informe a quantidade de pessoas que serão entrevistadas: 6

Informe o salário e a localidade da pessoa 1: 3000.00 U

Informe o salário e a localidade da pessoa 2: 9800.00 U

Informe o salário e a localidade da pessoa 3: 2500.00 R

Informe o salário e a localidade da pessoa 4: 1500.00 U

Informe o salário e a localidade da pessoa 5: 12100.00 R

Informe o salário e a localidade da pessoa 6: 2200.00 R

Maior renda mensal: R\$ 12100.00

Menor renda mensal: R\$ 1500.00

Percentual de entrevistados de áreas urbanas: 50%

Percentual de entrevistados de áreas rurais: 50%

Média da renda nas áreas urbanas: R\$ 4766.67

Média da renda nas áreas rurais: R\$ 5600.00

3. (Desafio - Baseado em questão da maratona de programação 2016) Faça um programa que tente adivinhar o número inteiro N que o usuário está pensando, com base em pistas que ele oferece. As pistas, neste caso, são 4 números inteiros que o usuário informa: um divisor de N, um não divisor de N, um múltiplo de N e um não múltiplo de N. O programa deve mostrar todos os números que satisfazem esses critérios, e mostrar para o usuário. Caso nenhum número satisfaça os critérios, o programa deve informar que não há números que satisfazem esses critérios.

Exemplo de execução:

Informe um divisor do seu número: 2

Informe um não divisor do seu número: 4

Informe um múltiplo do seu número: 90

Informe um não múltiplo do seu número: 40

Seu número é um desses:

30

90

4. (Desafio)A fórmula de Leibniz para calcular a constante π , envolve a seguinte série infinita:

$$\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots = \frac{\pi}{4}$$

Quanto maior for o número de termos considerados, maior será a precisão da aproximação. Faça um programa que lê um valor N que representa o número de termos e que calcula o valor aproximado de π , utilizando o número de termos informado.

5. Faça um programa que recebe dois números inteiros entre 0 e 19 que representam as coordenadas x e y de um ponto em um plano bidimensional 20X20, cuja origem fica no canto inferior esquerdo. A função deve desenhar este plano na tela, usando caracteres “#”, e indicando a posição do ponto com o caractere “P”. Por exemplo, se o ponto informado fosse (5,8), a função deveria desenhar o seguinte (ignore a cor diferenciada):

```
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####P#####
#####
#####
#####
#####
#####
#####
#####
#####
```

6. (Desafio) Barras de progresso constituem uma maneira de informar o usuário sobre a quantidade de informação processada até um certo momento. Dado um número real x entre 0.0 e 1.0, que representa fração da quantidade informação processada, e um número inteiro N entre 5 e 50, que representa o tamanho da barra (em caracteres), faça um programa que leia x e N, e imprima a barra de progresso. A barra deve ser impressa de acordo com as regras abaixo:
- “<” indica o início da barra
 - “#” indica informação processada
 - “-” indica informação faltante
 - “>” indica o final da barra

Exemplo de execução

```
Entre com o progresso atual (x) e tamanho (N) da barra: 0.50 10
<#####----->
Entre com o progresso atual (x) e tamanho (N) da barra: 0.75 15
<#####----->
```