

# IFRS – Campus Porto Alegre

Disciplina: Linguagem de Programação I - 2016/2 29/09/2016

Prof: Fabio Okuyama

Envie arquivos fonte em um único arquivo ZIP

Após o envio, confira o arquivo enviado, baixando-o do moodle.

Arquivos ZIP estragados receberão nota zero.

Não é permitido usar chat/email e outras formas de comunicação.

Nota: Os três programas devem conter comentários e devem imprimir na tela o nome do programador (ex. "Programador: Fabio")

**Questão 1: Faça um programa em linguagem C que execute as seguintes tarefas:**

- Peça ao usuário números inteiros positivos, deve parar de pedir números quando o usuário digitar '0' (zero) Nesta tarefa é obrigatório o uso correto de uma estrutura de repetição (for, while ou do.. while);
- Para cada número inteiro positivo digitado pelo usuário, o programa deve mostrar os divisores dos valores ímpares e a tabuada dos valores pares;
- Quando o laço de repetição terminar (isso ocorrerá quando o usuário digitar '0'), o programa deverá mostrar ao usuário a **Média de todos dos valores ímpares** e o **Maior valor par digitado pelo usuário**: Desconsiderar o valor '0' (zero) para todos os itens.

## SUGESTÃO:

- Identifique no enunciado acima o que deve ser pedido ao usuário:
- Que variáveis você irá utilizar no programa? Qual tipo?
- Identifique no enunciado acima o que o programa deve fazer:
- Faça um esboço do seu programa em formato de fluxograma:

**Teste:**

**Digamos que o usuário digitou os seguintes valores:**

9	7	2	5	6	0
---	---	---	---	---	---

**O programa deverá apresentar para o usuário:**

Digite um número inteiro positivo: 9

Divisores do nº9: 1, 3, 9,

Digite um número inteiro positivo: 7

Divisores do nº7: 1, 7,

Digite um número inteiro positivo: 2

Tabuada do nº2:

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$2 \times 4 = 8$$

$$2 \times 5 = 10$$

$$2 \times 6 = 12$$

$$2 \times 7 = 14$$

$$2 \times 8 = 16$$

$$2 \times 9 = 18$$

$$2 \times 10 = 20$$

Digite um número inteiro positivo: 5

Divisores do nº5: 1, 5,

Digite um número inteiro positivo: 6

Tabuada do nº6:

$$6 \times 1 = 6$$

$$6 \times 2 = 12$$

$$6 \times 3 = 18$$

$$6 \times 4 = 24$$

$$6 \times 5 = 30$$

$$6 \times 6 = 36$$

$$6 \times 7 = 42$$

$$6 \times 8 = 48$$

$$6 \times 9 = 54$$

$$6 \times 10 = 60$$

Digite um número inteiro positivo: 0

Bye-bye

**Questão 2: Cálculo de Investimento: O usuário deve informar:**

- i) o percentual de rendimento mensal de uma aplicação;**
- ii) o valor mensal de depósito;**
- iii) Meta a ser atingida (Valor objetivo).**

**Calcular em quantos meses a meta será atingida**

Por exemplo, se o usuário informar que o rendimento mensal é 0,5%; o valor de depósito mensal é de R\$100,00; meta a ser atingida é 1000, (1 mil reais) teremos:

<b>Mes 1</b>	100	
<b>Mes 2</b>	200,5	0,50 de juros + 100 de depósito
<b>Mes 3</b>	301,5025	1,0025 de juros + 100 de depósito
<b>Mes 4</b>	403,0100125	1,5075125 de juros + 100 de depósito
<b>Mes 5</b>	505,0250625625	2,0150500625 de juros + 100 de depósito
<b>Mes 6</b>	607,550187875312	2,5251253128125 de juros + 100 de depósito
<b>Mes 7</b>	710,587938814689	3,03775093937656 de juros + 100 de depósito
<b>Mes 8</b>	814,140878508762	3,55293969407344 de juros + 100 de depósito
<b>Mes 9</b>	918,211582901306	4,07070439254381 de juros + 100 de depósito
<b>Mes 10</b>	1022,80264081581	4,59105791450653 de juros + 100 de depósito

No exemplo acima, o programa deve apresentar até o mes 10, onde será alcançado valor de 1000 reais.

Não é necessário decompor o resultado.

**a - Leitura correta dos dados**

**b - Uso correto de estrutura de Repetição**

**c - Apresentar o saldo a cada mês até que o saldo seja maior que a meta.**

Nota: o Programa deve conter comentários e o nome do autor.

Ao FINAL deve ser exibido o nome do programador.

**Questão 3: - Em uma eleição existem 4 candidatos à vereador e 3 candidatos à prefeitura. O programa deve realizar a leitura do voto para vereador e para prefeito.**

Os votos são informados através de códigos.

Os dados utilizados para a contagem dos votos obedecem à seguinte codificação:

Para vereador os códigos são: 66666 para candidato A; 77777 para candidato B; 88888 para candidato C; 99999 para candidato D

Para prefeito os códigos são: 99 para candidato X , 88 para Candidato Y; 77 para candidato Z;

111 = voto em branco (para ambas as votações);

-Outro valor: Voto Nulo;

Elabore um algoritmo que realize a leitura de ambos os votos (vereador e depois para prefeito) leia os códigos do candidato que receberá o voto. **O programa deverá ler 7 votos (7 para vereador e 7 para prefeito) alternadamente.**

**Calcule e escreva:**

-total de votos para cada candidato e seu respectivo percentual de votos sobre os votos válidos\*; (em cada uma das duas eleições)

-total de votos nulos; (em cada uma das duas eleições)

-total de votos em branco; (em cada uma das duas eleições)

**Votos válidos são todos os votos - nulos.**

**Lembre-se: os votos em branco e nulos podem ser diferentes nas duas eleições.**

a - Leitura correta dos dados;

b - Estrutura de repetição ;

c - Cálculo correto da Contagem de Votos ;

Nota: o Programa deve conter comentários e o nome do autor.

Ao FINAL deve ser exibido o nome do programador.

### **Guia de Correção, para cada Exercício:**

Funcionalidades a,b,c: 66,6% da nota (22,2% cada item)

Critério 1: 22,2% da nota

Critério 5: 11,1% da nota

#### **Critério 2 - Peso 1**

- Funcionalidade **a** implementada corretamente e funcional (100%)
- Funcionalidade **a** implementada com erro facilmente corrigível(75%)
- Funcionalidade **a** implementada parcialmente (25%)
- Funcionalidade **a** não implementada ou com erro(s) grave(s) (0%)

#### **Critério 3 - Peso 1**

- Funcionalidade **b** implementada corretamente e funcional (100%)
- Funcionalidade **b** implementada com erro facilmente corrigível(75%)
- Funcionalidade **b** implementada parcialmente (25%)
- Funcionalidade **b** não implementada ou com erro(s) grave(s) (0%)

#### **Critério 4 - Peso 1**

- Funcionalidade **c** implementada corretamente e funcional (100%)
- Funcionalidade **c** implementada com erro facilmente corrigível(75%)
- Funcionalidade **c** implementada parcialmente (25%)
- Funcionalidade **c** não implementada ou com erro(s) grave(s) (0%)

#### **Critério 1 (Peso 1):**

- Programador apresentou domínio sobre o código, realizando a apresentação do funcionamento do mesmo, mostrando onde cada funcionalidade foi implementada (100%)
- Programador apresentou domínio sobre o código, mas tem poucas dúvidas sobre o funcionamento do mesmo e onde estão implementadas as funcionalidade (75%)
- Programador fez o código mas não compreende seu funcionamento e tem dificuldades para dizer onde estão implementadas as funcionalidades (25%)
- Exercício não implementado (0%)

#### **Critério 5 (peso 0,5)**

- Programa funcionando corretamente (100%)
- Programa não funciona com erros facilmente corrigíveis(50%)
- Programa não implementado ou com erro(s) grave(s) (0%)