



## Laboratório 04

### Exercícios Resolvidos

#### Exercícios :

**Exercício 01:** Faça um algoritmo para calcular a área de uma circunferência, considerando a fórmula  $AREA = \pi * RAI^2$ . Utilize uma função para esse cálculo, sendo que o algoritmo principal irá apenas obter os valores das variáveis e apresentar o resultado final.

funcao *calculaAreaCircunferencia ( raio : real ) : real*

var

*area : real*

inicio

*area*  $\leftarrow \pi * ( raio ^ 2 )$

**retorne** *area*

fimfuncao

algoritmo "Ex01"

var

*area : real*

*raio : real*

inicio

**escreva** ( " Informe o raio da circunferência em cm: " )

**leia** ( *raio* )

*area*  $\leftarrow$  *calculaAreaCircunferencia ( raio )*

```
escreval ( )  
  
escreval ( )  
  
escreva ( " A area da circunferência de raio ", raio , "cm é : ", area , "cm²." )  
  
escreval ( )
```

[finalgoritmo](#)

**Exercício 02:** Faça um algoritmo que:

- a) Obtenha o valor para a variável hrsTrabalhadas (horas trabalhadas no mês)
- b) Obtenha o valor para a variável valorHora (valor hora trabalhada)
- c) Obtenha o valor para a variável percentDesconto (percentual de desconto)
- d) Calcule o salário bruto =>  $\text{salarioBruto} = \text{hrsTrabalhadas} * \text{valorHora}$
- e) Calcule o total de desconto =>  $\text{totalDesconto} = (\text{percentDesconto}/100) * \text{salarioBruto}$
- f) Calcule o salário líquido =>  $\text{salarioLiquido} = \text{salarioBruto} - \text{totalDesconto}$
- g) Apresente os valores de: Horas trabalhadas, Salário Bruto, Desconto, Salário Líquido.

Utilize um procedimento para realizar este cálculo, deixando o algoritmo principal apenas os dados.

[procedimento](#) *calculaValores ( hrsTrabalhadas, valorHora, percentDesconto : real )*

[var](#)

```
salarioBruto : real  
  
totalDesconto : real  
  
salarioLiquido : real
```

[inicio](#)

```
salarioBruto ← hrsTrabalhadas * valorHora  
  
totalDesconto ← (percentDesconto / 100 ) * salarioBruto  
  
salarioLiquido ← salarioBruto - totalDesconto  
  
escreval ( )  
  
escreval ( )
```

*escreval ( " Horas trabalhadas : ", hrsTrabalhadas, " horas" )*

*escreval ( " Salário Bruto : R\$", salarioBruto , ",00" )*

*escreval ( " Desconto : R\$", totalDesconto, ",00" )*

*escreval ( " Salário Líquido : R\$", salarioLiquido, ",00" )*

fimprocedimento

algoritmo "Ex02"

var

*hrsTrabalhadas : real*

*valorHora : real*

*percentDesconto : real*

inicio

*escreva ( " Informe a quantidade de horas trabalhadas : " )*

*leia ( hrsTrabalhadas )*

*escreva ( " Informe o valor da hora trabalhada : " )*

*leia( valorHora )*

*escreva ( " Informe a porcentagem do desconto ( Ex.: 10% = 10 ): " )*

*leia ( percentDesconto )*

*calculaValores( hrsTrabalhadas, valorHora, percentDesconto )*

fimalgoritmo

**Exercício 03:** Construir um algoritmo que tome como entrada dois valores inteiros distintos e os apresente (imprima) em ordem crescente (menor para o maior). Essa validação deverá ser feita por uma função, que compara dois números inteiros e retorna o menor deles. Utilize um vetor na solução do problema.

funcao *comparaInteiros( x, y : inteiro ) : inteiro*

var

*menor : inteiro*

inicio

*se x < y entao*

*menor  $\leftarrow$  x*

*senao*

*menor  $\leftarrow$  y*

*fimse*

*retorne menor*

fimfuncao

algoritmo "Ex03"

var

*num1, num2, indice : inteiro*

*vetCrescente : vetor[0..1] de inteiro*

inicio

*escreva ( " Informe o primeiro valor : " )*

*leia (num1)*

*escreva ( " Informe o segundo valor : " )*

*leia (num2)*

*se comparaInteiros(num1, num2) = num1 entao*

*vetCrescente[0]  $\leftarrow$  num1*

*vetCrescente[1]  $\leftarrow$  num2*

*senao*

*vetCrescente[0]  $\leftarrow$  num2*

```
vetCrescente[1] ← num1
```

```
fimse
```

```
escreval ( )
```

```
escreva ( " Ordem Crescente : " )
```

```
para indice de 0 ate 1 faca
```

```
escreva ( vetCrescente[indice] , " " )
```

```
fimpara
```

```
escreval ( )
```

```
fimalgoritmo
```

**Exercício 04:** Um hotel cobra R\$ 60.00 a diária e mais uma taxa de serviços. A taxa de serviços é de:

- R\$ 5.50 por diária, se o número de diárias for maior que 15.
- R\$ 6.00 por diária, se o número de diárias for igual a 15.
- R\$ 8.00 por diária, se o número de diárias for menor que 15.

Construa uma função calcule o valor a ser pago pelo cliente após sua estadia. Construa um algoritmo que obtenha o nome e a quantidade de diárias que este usufruiu, apresentando seu nome e o total da conta do cliente. Utilize a função para esse cálculo.

```
funcao calculaContaCliente ( qtdDiarias : inteiro ) : real
```

```
var
```

```
valorConta : real
```

```
inicio
```

```
se qtdDiarias > 15 entao
```

```
valorConta ← 60 + ( 5.50 * qtdDiarias )
```

```
senao
```

```
se qtdDiarias < 15 entao
```

```
valorConta ← 60 + ( 6 * qtdDiarias )
```

*senao*

$\text{valorConta} \leftarrow 60 + (8 * \text{qtdDiarias})$

*fimse*

*fimse*

*retorne valorConta*

fimfuncao

algoritmo "Ex04"

var

*nome : literal*

*qtdDiarias : inteiro*

*conta : real*

inicio

*escreva ( " Informe o nome do cliente : " )*

*leia (nome)*

*escreva ( " Informe a quantidade de diárias consumidas " )*

*leia (qtdDiarias)*

*conta  $\leftarrow$  calculaContaCliente ( qtdDiarias )*

*escreval ( )*

*escreva ( " Total a ser pago pelo cliente ", nome , " é : R\$ " , conta , ",00 " )*

*escreval ( )*

fimalgoritmo

**Exercício 05:** O cardápio de uma lanchonete é o seguinte:

Especificação	Preço unitário
100 - Cachorro quente	R\$ 1,10
101 - Bauru simples	R\$ 1,30
102 - Bauru c/ovo	R\$ 1,50
103 - Hamburger	R\$ 1,10
104 - Cheeseburger	R\$ 1,30
105 - Refrigerante	R\$ 1,00

Escrever um algoritmo que leia o código dos itens pedidos, a quantidade e calcule o valor a ser pago por aquele lanche. A condição de parada que indica o fim dos pedidos é o código **0**.

Crie um algoritmo modularizado.

funcao *getPrecoItem ( codigo : inteiro ) : real*

var

*preco : real*

inicio

*escolha codigo*

*caso 100 , 103*

*preco ← 1.10*

*caso 101 , 104*

*preco ← 1.30*

*caso 102*

*preco ← 1.5*

*caso 105*

*preco ← 1*

*outrocaso*

*preco ← -1*

*fimescolha*

*retorne preco*

fimfuncao

algoritmo "Ex05"

var

*conta, precoItem : real*

*codigo, quantidade : inteiro*

inicio

*escreval ( "Informe os códigos dos itens pedidos. " )*

*escreval ( "Digite código 0 para finalizar o cálculo. " )*

*escreval ( )*

*repita*

*escreva ( " Informe o código do produto : " )*

*leia ( codigo )*

*precoItem ← getPrecoItem ( codigo )*

*se precoItem <> -1 ) entao*

*escreva ( " Informe a quantidade consumida : " )*

*leia ( quantidade )*

*conta ← conta + ( precoItem \* quantidade ) )*

*limpatela*

*escreval ( " Continue calculando a conta... " )*

*escreval ( )*

*senao*

*escreval ( "Codigo inválido!" )*

*fimse*



```
ate codigo = 0
```

```
escreval ( )
```

```
escreval ( " Total da conta : R$", conta )
```

fimalgoritmo

**Exercício 06:** Construa um algoritmo pelo qual um usuário possa escolher que tipo de média deseja calcular a partir de três notas. O algoritmo deve ler as notas, a opção escolhida pelo usuário e invocar uma das duas funções que realiza o cálculo da média, de acordo com a opção selecionada:

1- aritmética

2- ponderada (pesos 3, 3, 4)

As funções irão retornar o valor final da média, para que o algoritmo principal apresente este ao usuário.

funcao *calcularMediaAritmetica* ( *nota1*, *nota2*, *nota3* : *real* ) : *real*

var

*media* : *real*

inicio

*media*  $\leftarrow$  ( *nota1* + *nota2* + *nota3* ) / 3

*retorne media*

fimfuncao

funcao *calcularMediaPonderada* ( *nota1*, *nota2*, *nota3* : *real* ) : *real*

var

*media* : *real*

inicio

*media*  $\leftarrow$  ( ( 3 \* *nota1* ) + ( 3 \* *nota2* ) + ( 4 \* *nota3* ) ) / ( 3 + 3 + 4 )

*retorne media*

fimfuncao

algoritmo "Ex06"

var

*opcao : inteiro*

*media, n1, n2, n3 : real*

inicio

**repita**

*escreva ( " Informe a primeira nota do aluno: " )*

*leia ( n1 )*

*escreva ( " Informe a segunda nota do aluno: " )*

*leia ( n2 )*

*escreva ( " Informe a terceira nota do aluno: " )*

*leia ( n3 )*

*escreval ( )*

*escreval ( )*

*escreval ( " Informe o tipo de média que deverá ser calculada : " )*

*escreval ( " 1 - aritmética " )*

*escreval ( " 2 - ponderada " )*

*leia ( opcao )*

*se opcao = 1 entao*

*media ← calcularMediaAritmetica ( n1, n2, n3 )*

**interrompa**

**senao**

*se opcao = 2 entao*

*media ← calcularMediaPonderada ( n1, n2, n3 )*

**interrompa**

**senao**

*escreval ( )*

*escreval ( )*

*escreval ( " Opção inválida " )*

*fimse*

*fimse*

*ate falso*

*escreval ( )*

*escreval ( )*

*escreval ( " Média do aluno : ", media )*

finalgoritmo

**Exercício 07:** Escreva um algoritmo que leia um número *n* (número de termos de uma progressão aritmética), *a1* (o primeiro termo da progressão) e *r* (a razão da progressão) e escreva os *n* termos desta progressão, bem como a soma dos elementos.

Construa um algoritmo modularizado.

procedimento *montarProgressaoAritmetica ( n : inteiro ; a1, r : real )*

var

*valor, somaElementos : real*

*cont : inteiro*

inicio

*valor ← a1*

*para cont de 1 ate n faca*

*escreva ( valor , " " )*

*somaElementos ← somaElementos + valor*

*valor ← valor + r*

*fimpara*

fimprocedimento

algoritmo "Ex07"

var

*n: inteiro*

*a1, r: real*

inicio

*escreva ( " Informe o número de elementos na progressão : " )*

*leia ( n )*

*escreva ( " Informe o valor do primeiro elemento : " )*

*leia ( a1 )*

*escreva ( " Informe a razão da progressão : " )*

*leia ( r )*

*escreval ( )*

*escreval ( )*

*escreval ( " Progressão aritmética formada: " )*

*escreval ( )*

*montarProgressaoAritmetica ( n, a1, r )*

*escreval ( )*

fimalgoritmo

**Exercício 08:** Construa um algoritmo que verifique se um número inteiro, passado como parâmetro para uma função, é par ; retornando verdadeiro em caso positivo e falso caso seja ímpar.

funcao *verificaPar ( n : inteiro ) : logico*

var

inicio

*se  $n \bmod 2 = 0$  entao*

*retorne verdadeiro*

*senao*

*retorne falso*

*fimse*

fimfuncao

algoritmo "Ex08"

var

*retorno : logico*

*n : inteiro*

inicio

*escreva ( " Informe um número : " )*

*leia ( n )*

*retorno  $\leftarrow$  verificaPar ( n )*

*se retorno entao*

*escreva ( " O número informado é par. " )*

*senao*

*escreva ( " O número informado é ímpar. " )*

*fimse*

*escreval ( )*

fimalgoritmo

**DESAFIO:**

Um determinado material radioativo perde metade de sua massa a cada 50 segundos. Dada a massa inicial, em gramas, faça um programa que determine o tempo necessário para que essa massa se torne menor que 0,05 gramas.

algoritmo "Desafio"

var

*valorInicial : real*

*tempo : inteiro*

inicio

*escreva ( " Informe o peso inicial do material em gramas : " )*

*leia ( valorInicial )*

*repita*

*valorInicial  $\leftarrow$  valorInicial \* ( 1 / 2 )*

*tempo  $\leftarrow$  tempo + 50*

*ate valorInicial  $\leq$  0.05*

*escreval ( )*

*escreval ( )*

*escreval ( " O tempo gasto para este material ficar com menos " )*

*escreval ( " de 0.05 gramas é", tempo, " segundos ou " )*

*escreval ( tempo / 60 , " minutos. " )*

*escreval ( )*

*escreval ( )*

fimalgoritmo