
Lista de Exercícios 01 (EXE01)

1. Numa construção **if** sem **else**, o que acontece se a condição de teste for falsa?
 - a. O controle procura pelo ultimo **else** do programa;
 - b. Nada;
 - c. O controle passa para a instrução seguinte ao **if**;
 - d. O corpo do comando **if** é executado.
2. Faça um programa que receba o número de lados de um polígono convexo ($N \geq 3$), calcule e mostre o número de **diagonais** desse polígono. Sendo N o número de lados do polígono, o número de diagonais é dado por: **$ND = N(N-3)/2$** . Obs.: O número de diagonais é sempre inteiro e caso N seja menor que 3, solicite novo valor para N até que N seja maior ou igual a 3.
3. Uma empresa determinou um **reajuste salarial** de 5% a todos os seus funcionários. Além disto, concedeu um **abono** de R\$ 100,00 para aqueles que recebem até R\$750,00. Dado o valor do salário de um funcionário, informar para quanto ele será reajustado.
4. Dados **2 números** imprimi-los em **ordem crescente**. Ordem crescente quando um é menor ou igual ao seguinte.
5. Dados **3 números** imprimir o **maior**.
6. Dados **3 números** imprimi-los em **ordem crescente**. Ordem crescente quando um é menor ou igual ao seguinte.
7. Use o operador condicional ternário para imprimir o **maior entre 2 números**.
8. Seja **e** uma variável inteira contendo o **número de erros** detectados em determinado processo. Use o **operador condicional ternário** para codificar uma instrução capaz de exibir saídas como:
1 erro detectado.
5 erros detectados.
9. Usando o operador condicional ternário, codifique uma instrução para exibir valores lógicos como **true** e **false**. Para o **valor 0** deve aparecer **false** e para **qualquer outro**, **true**.
10. A vantagem de uma construção **switch** sobre um **if-else** é:
 - a. Uso do **default** no switch;
 - b. O switch fornece **clareza e facilidade de leitura**;
 - c. Os casos de um switch permitem **diversas escolhas**;
 - d. **Várias instruções** podem ser executadas em cada caso de um switch.
11. Verdadeiro ou Falso.
 - a. Toda construção **switch** pode ser transformado em **if encadeado**.
 - b. Todo **if encadeado** pode ser transformado em uma construção **switch**.
12. Usando a **estrutura de decisão múltipla**, codifique um programa que leia os 4 dígitos da placa de um carro ($0 < \text{placa} \leq 9999$) e informe o dia do seu **rodízio** (dica: use o operador % para obter o ultimo dígito da placa).

Lista de Exercícios 01 (EXE01)

13. Um comando break:
- Termina um **programa**;
 - É obrigatório em uma construção **switch**;
 - Causa saída imediata de um **if**;
 - Causa saída imediata de um **laço (for; while; do-while)**.
14. Quais os valores das variáveis x e y após a execução dessas instruções? Por quê?
- Seja **x=5** e a instrução **y = x++ + ++x**.
 - Seja **x=3** e a instrução **y = x * (x + 1) * x++**.
 - Seja **x=5, y = 3** e a instrução **y *= x+1**.
15. As três **expressões** que compõem a expressão do laço **for** são **separadas por** _____.
16. Um laço **do-while** é útil quando seu corpo:
- Nunca é executado;
 - Pode nunca ser executado;
 - Deve ser executado pelo menos uma vez;
 - Termina após a primeira execução.
17. Dado **n** positivo e uma **seqüência de n números** calcular a **soma** dos elementos da seqüência. Faça um loop para validar se n é maior que zero, ou seja, enquanto $n \leq 0$ exiba uma mensagem de erro e solicite um novo valor de n.
18. Idem anterior, calculando a **média**.
19. Idem anterior, determinando o **maior** e o **menor**.
20. Dado $N \geq 0$ inteiro e X real, calcular X elevado a **potência N** (sem usar pow).
21. Dado $N \geq 0$ inteiro, calcular **N!** (N fatorial)
22. Dado $N \geq 0$ inteiro e X real, calcular o valor da **soma**:
- $$1 + X + X^2 + \dots + X^N$$
23. Um comerciante precisa informatizar o caixa de sua loja. Para isso ele precisa de um programa que leia uma série de valores correspondendo aos **preços das mercadorias** compradas por um cliente (o **valor zero** finaliza a entrada), calcule o **valor total**, subtraia deste valor o **desconto** devido (vide tabela) e, finalmente, mostre o **valor a ser pago** pelo cliente. Codifique esse programa.

<i>Total</i>	<i>Desconto</i>
abaixo de R\$ 50,00	5%
até R\$ 100,00	10%
até R\$ 200,00	15%
acima de R\$ 200,00	20%

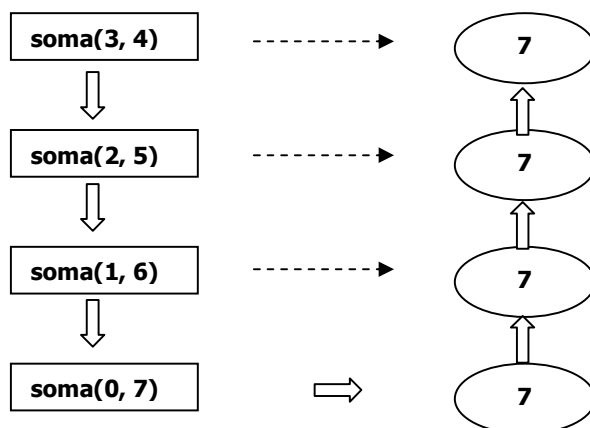
Lista de Exercícios 01 (EXE01)

24. Escreva uma função recursiva que calcule a somatória de n termos, sendo n um número natural ($n \geq 0$). A função $somatoria(n)$ é definida da seguinte maneira:

$$somatoria(n) = \begin{cases} 0 & \text{se } n = 0 \\ n + \dots + 3 + 2 + 1 & \text{se } n > 0 \end{cases}$$

25. Usando apenas subtração, crie uma função recursiva para determinar se um número natural n é par ou não.
26. Crie uma função recursiva para calcular a soma dos n primeiros termos da série harmônica, ou seja, $1 + 1/2 + 1/3 + \dots + 1/n$. Cuidado, a divisão de dois números inteiros tem como resultado um inteiro, ou seja, para que a soma seja calculada corretamente, sugiro que utilize: $1.0 + 1.0/2 + 1.0/3 + \dots + 1.0/n$
27. Escreva uma função recursiva para calcular a soma de dois números naturais n e m usando as funções **pred(i)** e **succ(i)** que devolvem respectivamente o **predecessor** e o **sucessor** de um inteiro i . Se $m = 0$, a soma de m com n é trivial e sua solução é n . Porém, se $m > 0$, a soma de m com n é igual a soma do **predecessor de m** com o **sucessor de n** .

Exemplo:



28. Considerando a função **soma(m, n)** definida na questão anterior, simule a execução das chamadas **soma(3,6)** e **soma(6,3)**. Em seguida altere esta função de modo que o número de passos executados por ela seja no máximo **menor (m, n) + 1**.
29. Codifique uma função recursiva para:
- Calcular o fatorial de um número natural.
 - Calcular o resto da divisão inteira usando subtração.
 - Calcular o quociente da divisão inteira usando subtração.
 - Calcular o produto de dois naturais usando adição.
 - Exibir uma contagem regressiva a partir de um valor n .
 - Exibir um número natural em binário.
 - Exibir um número natural invertido (da direita para esquerda).