



Lista 3 - Estrutura de Repetição

Exercícios de Sala de Aula

Observações para os exercícios desta lista:

a) Fazer os programas de forma que o usuário possa optar por repetir a execução dos mesmos, ou seja, ao final da execução, questionar se o usuário deseja continuar utilizando o programa. Se informado afirmativamente, repetir a execução. Isso pode ser feito utilizando uma estrutura **do - while** que conterà a parte do programa que será repetida. Atenção para a inicialização das variáveis: deve ser feita dentro dessa estrutura de repetição.

b) Validar as entradas.

c) Validar para que não sejam realizadas divisões por zero.

1) Ler um número positivo, validar a entrada repetindo a leitura até que seja informado um número que atende essa condição. Esse número representa a quantidade de números primos a serem mostrados.

Exemplo:

```
Informe a quantidade de numeros primos que serao mostrados: 15
2      3      5      7      11     13     17     19     23
29     31     37     41     43     47
Deseja digitar outro numero (S/s): n
```

2) Ler um número positivo, validar a entrada repetindo a leitura até que seja informado um número que atende essa condição. Esse número representa a quantidade de números ímpares a serem mostrados. Apresentar esses valores n por linha, sendo n um número maior que zero, informado pelo usuário. Os valores são apresentados separados por tabulação. Fazer a média dos números ímpares mostrados.

Exemplo:

```
Informe o numero de impares que deseja mostrar: 15
Quantos impares por linha deseja mostrar: 5
 1      3      5      7      9
11     13     15     17     19
21     23     25     27     29

A media dos impares eh: 15.00

Deseja repetir a execucao do programa (S ou N): n
```

3) Escreva um programa que imprima todas as possibilidades de que no lançamento de dois dados a soma das faces seja um valor informado pelo usuário. O usuário deve informar um valor válido entre 2 a 12. Repetir a leitura até que o usuário informe um valor válido.

Exemplo:

```
Informe um valor entre 2 a 12: 7
1 + 6 = 7
2 + 5 = 7
3 + 4 = 7
4 + 3 = 7
5 + 2 = 7
6 + 1 = 7

Numero de possibilidades: 6

Deseja continuar no programa (S ou N): n
```

4) Escreva um programa que leia o valor de uma aplicação, o percentual de rendimento mensal obtido por esta aplicação e o período do investimento; e retorne o valor da aplicação ao final do período de investimento. Obs.: A cada 12 meses o percentual de rendimento deve ser aumentado em 0,25%. Validar o valor da aplicação para que seja positivo. Validar o percentual de rendimento para que seja um número entre 0 e 1. Validar o período para que seja um valor positivo.

Exemplo:

```
Informe o valor da aplicacao: R$ 1000
Informe o percentual de rendimento mensal (0 a 1): 0.05
Informe o periodo (em meses): 14
```

MES	% DE RENDIMENTO	VALOR APLICACAO
1	0.05	R\$ 1050.00
2	0.05	R\$ 1102.50
3	0.05	R\$ 1157.63
4	0.05	R\$ 1215.51
5	0.05	R\$ 1276.28
6	0.05	R\$ 1340.10
7	0.05	R\$ 1407.10
8	0.05	R\$ 1477.46
9	0.05	R\$ 1551.33
10	0.05	R\$ 1628.89
11	0.05	R\$ 1710.34
12	0.30	R\$ 2223.44
13	0.30	R\$ 2890.47
14	0.30	R\$ 3757.62

```
Deseja continuar no programa (S ou N): n
```

5) Apresentar os n primeiros números pares e divisíveis por 3 informados pelo usuário (validar para que n seja diferente de 0). Se o usuário informar um número negativo transformá-lo em positivo (sem usar funções prontas) antes de utilizar.

Por exemplo, se o usuário informar 4, o programa deverá mostrar os quatro primeiros números pares que são divisíveis por três: 0, 6, 12 e 18.

Exemplo:

```
Quantos numeros pares divisiveis por 3 quer mostrar? 4
0      6      12      18

Deseja continuar no programa (S ou N): s

Quantos numeros pares divisiveis por 3 quer mostrar? -4
0      6      12      18

Deseja continuar no programa (S ou N):
```

6) O Triângulo de Floyd é um triângulo formado com números naturais. O triângulo começa em 1 no canto superior esquerdo e continua a escrever a sequência de números naturais de tal modo que cada linha contém um número a mais do que a linha anterior. Escreva um programa que leia um número inteiro positivo n e em seguida imprima n linhas do Triângulo de Floyd.

Exemplo:

```
Informe um numero positivo: 10
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55
```

7) Fazer um programa que calcule o fatorial dos números inteiros de 1 até 12. Fazer com que a saída do programa seja mostrada como no exemplo a seguir:

Exemplo:

```
1 ! => 1 = 1
2 ! => 2 * 1 = 2
3 ! => 3 * 2 * 1 = 6
4 ! => 4 * 3 * 2 * 1 = 24
5 ! => 5 * 4 * 3 * 2 * 1 = 120
6 ! => 6 * 5 * 4 * 3 * 2 * 1 = 720
7 ! => 7 * 6 * 5 * 4 * 3 * 2 * 1 = 5040
8 ! => 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 40320
9 ! => 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 362880
10 ! => 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 3628800
11 ! => 11 * 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 39916800
12 ! => 12 * 11 * 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 479001600
```

8) Fazer um programa para mostrar os divisores, calcular a quantidade deles e mostrar essa quantidade para os números compreendidos entre o valor 'x' (informado pelo usuário) e 'x+10', inclusive. Validar a entrada, o usuário deverá fornecer um número positivo entre 2 e 100. Ao final, mostrar a maior quantidade de divisores.

A seguir um exemplo da execução, utilizá-lo como modelo para o programa implementado:

```
Informe um numero entre 2 e 100: 2
2 ==> 1, 2, 2 divisores.
3 ==> 1, 3, 2 divisores.
4 ==> 1, 2, 4, 3 divisores.
5 ==> 1, 5, 2 divisores.
6 ==> 1, 2, 3, 6, 4 divisores.
7 ==> 1, 7, 2 divisores.
8 ==> 1, 2, 4, 8, 4 divisores.
9 ==> 1, 3, 9, 3 divisores.
10 ==> 1, 2, 5, 10, 4 divisores.
11 ==> 1, 11, 2 divisores.
12 ==> 1, 2, 3, 4, 6, 12, 6 divisores.

A maior quantidade de divisores eh: 6

Deseja continuar no programa (S ou N)? n
```

9) Construa um programa que leia um conjunto de dados contendo altura (que deve ser maior que 0) e sexo (que deve ser masculino ou feminino) de um grupo de pessoas. Quando fornecida uma altura igual a 0, é encerrada a entrada de dados. O programa deve calcular e fornecer:

- a) A maior e a menor altura do grupo.
- b) A média de altura das mulheres.
- c) A média de altura dos homens.
- d) O número de homens e o percentual que eles representam no grupo.
- e) O número de mulheres e o percentual que elas representam no grupo.

Exemplo:

```
Informe a altura: 1.7
Informe o sexo: f

Informe a altura: 1.8
Informe o sexo: m

Informe a altura: 1.75
Informe o sexo: f

Informe a altura: 0

A maior altura do grupo eh: 1.80
A menor altura do grupo eh: 1.70

==== MULHERES ====
Quantidade: 2
Media de altura: 1.73
Percentual: 66.67

===== HOMENS =====
Quantidade: 1
Media de altura: 1.80
Percentual: 33.33

Deseja continuar no programa (S ou N)?
```

10) Ler dois números que representam os limites de um intervalo. Validar a entrada, o usuário deverá fornecer valores positivos. Mostrar o intervalo em ordem crescente.

- a) Mostrar os números primos desse intervalo, com cinco números por linha.
- b) Calcular a média dos múltiplos de 3 e de 7 desse intervalo. Validar para que não seja realizada uma divisão por zero.

Exemplo:

```
Informe um valor para o limite inferior do intervalo: 1
Informe um valor para o limite superior do intervalo: 100
 2   3   5   7  11
13  17  19  23  29
31  37  41  43  47
53  59  61  67  71
73  79  83  89  97

Media dos multiplos de 3 e 7: 52.50
```

11) Sendo n um número positivo, apresentar as n primeiras raízes quadradas exatas. A função para obter a raiz quadrada é a `sqrt()` e está na biblioteca `math.h`.

Exemplo:

```
Informe o valor de n: 10
Raiz quadrada de 1 ==> 1
Raiz quadrada de 4 ==> 2
Raiz quadrada de 9 ==> 3
Raiz quadrada de 16 ==> 4
Raiz quadrada de 25 ==> 5
Raiz quadrada de 36 ==> 6
Raiz quadrada de 49 ==> 7
Raiz quadrada de 64 ==> 8
Raiz quadrada de 81 ==> 9
Raiz quadrada de 100 ==> 10
Deseja repetir o programa (S ou N)?
```