



## Lista 2 - Estrutura de Repetição

### Exercícios para sala de aula

1) Escreva um algoritmo que gere a série de *Fibonacci* até o termo  $n$  que é informado pelo usuário. A série de *Fibonacci* é formada pela sequência: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... , etc.

Exemplo:

```
Quantos termos mostrar? 10
1      1      2      3      5      8      13      21      34
55      89
```

2) Faça um programa que imprima os  $n$  (indicado pelo usuário) primeiros números pares (começa em 0). Apresente 5 valores por linha.

Exemplo:

```
Quantos numeros pares quer mostrar? 22
0      2      4      6      8
10     12     14     16     18
20     22     24     26     28
30     32     34     36     38
40     42
```

3) Uma árvore A tem 1,50 metros e cresce 2 centímetros por ano, enquanto uma árvore B tem 1,10 metros e cresce 3 centímetros por ano. Construa um programa que calcule e imprima quantos anos serão necessários para que a árvore B seja maior que a árvore A.

Exemplo:

```
Altura final da arvore A = 2.32
Altura final da arvore B = 2.33

Serao necessarios 41 anos para que a arvore B seja maior que a arvore
A.
```

4) Ler caracteres e parar quando o usuário informar o caractere '0'. Utilizando a tabela ASCII contar quantos caracteres são letras maiúsculas, quantos são letras minúsculas e quantos caracteres não são letras. Sugestão: utilizar o código numérico para comparar (A equivale a 65, B a 66...).

Exemplo:

```
Informe um caractere: D
Informe um caractere: b
Informe um caractere: 4
Informe um caractere: @
Informe um caractere: Y
Informe um caractere: a
Informe um caractere: *
Informe um caractere: 0

Foram informadas 2 letras maiusculas.
Foram informadas 2 letras minusculas.
Foram informados 3 caracteres que nao sao letras.
```

5) Elaborar um programa para automatizar o caixa de uma loja. Ler a quantidade e o valor unitário de cada produto. Calcular o valor total da compra. Informar esse valor para o usuário e o usuário informa o valor para pagamento. O programa calcula o troco e informa-o ao usuário em notas de 10, 5, 1 e os centavos. Prosseguir a leitura enquanto a quantidade for maior que zero. Se informado zero ou valor negativo para a quantidade, não ler o valor unitário.

Exemplo:

```
Informe a quantidade de um produto: 3
Informe o valor unitario do produto: 12.45
Informe a quantidade de um produto: 4
Informe o valor unitario do produto: 5.8
Informe a quantidade de um produto: 0
TOTAL DA COMPRA: R$ 60.55

Informe o valor pago pelo cliente: 100
TROCO: 39.45

O valor do troco sera devolvido com:
3 nota(s) de 10 reais
1 nota(s) de 5 reais
4 nota(s) de 1 real
e 0.45 centavos
```

6) Ler números inteiros. Prosseguir a leitura até ser informado o valor zero que não deve ser considerado nos cálculos. Informar:

- a) Quantos valores positivos foram informados.
- b) Quantos valores negativos e pares foram informados. Exemplo: Se a entrada for -1, -2, -3, deverá retornar 1, pois tem apenas um número negativo que é par.
- c) A média dos valores divisíveis por 3 informados. Usar a função fabs() para converter os valores negativos em positivos. Validar para não fazer divisão por zero no cálculo da média.

Exemplo:

```
Informe um numero: -1
Informe um numero: -2
Informe um numero: -3
Informe um numero: 1
Informe um numero: 2
Informe um numero: 3
Informe um numero: 0

Numeros positivos: 3

Numeros negativos que sao pares: 1

A media dos numeros divisiveis por 3 eh: 3.0
```

7) Elaborar um programa que determine a potência de um número. O usuário informa a base e o expoente e o programa informa o resultado da base elevada ao expoente. Não usar funções prontas do C para fazer o cálculo da potência.

Exemplo:

```
Informe o valor da base: 2
Informe o valor do expoente: 5
2 * 2 * 2 * 2 * 2 = 32
```

8) Elaborar um programa que valide o número de uma conta corrente com três dígitos, retorne o dígito verificador e informe se o número da conta está correto. Ler os quatro dígitos como um único número, validando para que esse número esteja em um intervalo de 1000 a 9999. Separar os dígitos de forma que os três primeiros sejam o número da conta e o último o dígito verificador. Validar a entrada, ou seja, repetir a entrada do número até que seja informado um número com somente quatro dígitos.

Por exemplo: Se o número informado for 1234, 123 é o número da conta e 4 é o dígito verificador.

Para obter o dígito verificador:

- Somar o número da conta com o seu inverso =  $123 + 321 = 444$
- Multiplicar cada dígito pela sua ordem posicional e somar os resultados =  $4 * 1 + 4 * 2 + 4 * 3 = 24$
- O último dígito desse resultado é o verificador: 4

Outro exemplo:

Se o usuário informar 1290:

- Somar o número da conta com o seu inverso =  $129 + 921 = 1050$
- Multiplicar cada dígito pela sua ordem posicional e somar os resultados =  $1 * 0 + 0 * 1 + 5 * 2 + 0 * 3 = 10$
- O último dígito desse resultado é o verificador: 0

Exemplo 1:

```
Informe o numero da conta corrente: 1234

CONTA: 1234
Conta corrente = 123

Primeiro digito = 1
Segundo digito = 2
Terceiro digito = 3
Quarto digito = 4

Inverso = 321
123 + 321 = 444

Primeiro digito = 0
Segundo digito = 4
Terceiro digito = 4
Quarto digito = 4

x = 24
Digito verificador = 4

O digito verificador esta correto!
```

Exemplo 2:

```
Informe o numero da conta corrente: 1290
```

```
CONTA: 1290
```

```
Conta corrente = 129
```

```
Primeiro digito = 1
```

```
Segundo digito = 2
```

```
Terceiro digito = 9
```

```
Quarto digito = 0
```

```
Inverso = 921
```

```
129 + 921 = 1050
```

```
Primeiro digito = 1
```

```
Segundo digito = 0
```

```
Terceiro digito = 5
```

```
Quarto digito = 0
```

```
x = 10
```

```
Digito verificador = 0
```

```
O digito verificador esta correto!
```

9) Implemente um programa que lê números informados pelo usuário. O algoritmo deve parar quando informado um número negativo. Para cada número lido, exceto o número negativo que representa a saída, o algoritmo deverá imprimir a soma dos dígitos.

Exemplo:

```
Digite um numero: 123
```

```
A soma dos digitos eh: 6
```

```
Digite um numero: 456
```

```
A soma dos digitos eh: 15
```

```
Digite um numero: 12
```

```
A soma dos digitos eh: 3
```

```
Digite um numero: 012
```

```
A soma dos digitos eh: 3
```

```
Digite um numero: 0
```

```
A soma dos digitos eh: 0
```

```
Digite um numero: -1
```

10) Um número abundante é aquele cuja soma dos seus divisores (excluído o próprio número) é maior do que ele mesmo. Por exemplo, o número 12 é abundante, pois se somarmos os seus divisores (1, 2, 3, 4 e 6) obteremos o valor 16 e este é maior do que o próprio número. Já o número 10 não é abundante, pois a somatória dos seus divisores (1, 2 e 5) é menor do que 10. Também o número 6 não é abundante, pois a soma de seus divisores (1, 2 e 3) é igual a 6. Escreva um programa que permita a leitura de um valor a partir do teclado e imprima uma mensagem informando se o número digitado é ou não abundante.

Exemplo 1:

```
Digite um numero: 12
```

```
Divisores:
```

```
1
```

```
2
```

```
3
```

```
4
```

```
6
```

```
-----
```

```
16
```

```
12 eh abundante porque a soma dos divisores eh maior que 12.
```

Exemplo 2:

```
Digite um numero: 10

Divisores:
1
2
5
-----
8

10 nao eh abundante porque a soma dos divisores eh menor que 10.
```

11) Faça o programa que apresente a seguinte saída como no exemplo, solicitando ao usuário o número máximo (deve ser positivo).

Exemplo:

```
Informe um numero positivo: 5
```

```
1  1  1  1  1
2  2  2  2  2
3  3  3  3  3
4  4  4  4  4
5  5  5  5  5
```

```
Deseja digitar outro numero (S ou N):
```

12) Faça o programa que apresente a seguinte saída como no exemplo, perguntando ao usuário o número máximo (deve ser positivo).

Exemplo:

```
Informe um numero positivo: 5
```

```
1
1  2
1  2  3
1  2  3  4
1  2  3  4  5
```

```
Deseja digitar outro numero (S ou N):
```