



## Lista 5 - Estrutura de Repetição

### Exercícios para sala de aula

1) Ler um número maior ou igual a 0 que representa a quantidade de valores pares que devem ser mostrados. Apresentar esses valores  $n$  por linha, sendo  $n$  um número positivo, informado pelo usuário. Os valores são apresentados separados por tabulação.

Exemplo:

```
Quantidade de pares a ser mostrada: 12
Quantidade de elementos por linha: 3
2      4      6
8      10     12
14     16     18
20     22     24
```

2) 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
```

3) Qual o objetivo do programa representado no código a seguir.

```
#include <stdio.h>

int main(void)
{
    int num, i;

    do
    {
        printf("Informe um numero entre 1 e 50: ");
        scanf("%d", &num);

        if(num < 1 || num > 50)
        {
            printf("Valor informado fora da faixa.\n");
        }
    } while(num < 1 || num > 50);

    for(i=num; i<=50; i=i+2)
```

```

{
    printf("\n%d\t", i);
}

return 0;
}

```

4) Elaborar um programa que solicita a idade de um profissional e se ele é do tipo autônomo ou funcionário, calcula e mostra:

- a) O total de profissionais autônomos com menos de 18 anos.
- b) A média das idades dos funcionários com mais de 18 anos.

O programa deve validar as entradas: permitir somente valores positivos (maior que 0) para a idade e A/a/F/f (autônomo ou funcionário) para o tipo de profissional.

O programa termina quando for informado 0 ou um valor negativo para a idade, que não deve ser considerado. Portanto, ao ser informado 0 ou um valor negativo para a idade, não deve ser solicitado o tipo de profissional.

5) 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!

6) (Desafio) Implemente um algoritmo 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 deve imprimir a quantidade de dígitos do número informado.

Observação: Desconsidere 0 como dígito se o mesmo estiver no início do número.

Exemplo:

Informe um numero: 012

O numero tem 2 digitos

Informe um numero: 12

O numero tem 2 digitos

Informe um numero: 102

O numero tem 3 digitos

Informe um numero: -1

7) Implemente um algoritmo 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: 18
A soma dos digitos eh: 9
```

8) 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.

9) Elaborar um programa que leia um número que representa a quantidade de valores ímpares e múltiplos de 5 que o usuário que ver. Se o usuário informar um valor negativo transformá-lo em positivo sem usar a função fabs(). Fazer a média dos valores múltiplos de 3 desse intervalo.

Exemplo:

Quantos numeros impares e multiplos de 5 quer mostrar: -5

5

15

25

35

45

A media dos numeros multiplos de 5 eh: 30.00