

Correção dos exercícios da aula anterior

- 1- Desenvolva um programa que leia um vetor com 10 posições de números inteiros. Em seguida, receba um novo valor do usuário e verifique se este valor se encontra no vetor.

1-

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int i, valor, encontrou;
    int numeros[15];
    for(i=1;i<=10;i++)//receber 10 números em cada posição do vetor
    {
        printf("Digite um numero: \n");
        scanf("%d",&numeros[i]);
    }
    printf("Digite um numero a ser verificado no vetor");
    scanf("%d", &valor);
    encontrou=0;
    for(i=1;i<=10;i++)//comparação do valor digitado com cada item do vetor
    {
        if(numeros[i]==valor)
        {
            encontrou=1; //variável de controle
        }
        if (encontrou==1)
        {
            printf("O valor se encontra do vetor\n");
        }
        else
        {
            printf("O valor nao se encontra do vetor\n");
        }
    }
    system("pause");
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Não veio nesta aula?

Procure compreender o Código desenvolvido e qualquer dúvida comente com o seu professor!

2- Com base na seguinte matriz:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

2.1- Faça um programa que some todas as linhas e todas as colunas e apresente o resultado de cada uma.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

► 2.1-

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int i=0, j=0; //i=linha j=coluna
    int mat [3][3]={1, 2, 3,
                    4, 5, 6,
                    7, 8, 9 };
    int linha=0, coluna =0; // Variáveis para acumular o valor de cada linha e de cada coluna

    printf(" Total por linha: \n\n");

    for(i=0;i<3;i++) //percorrer o vetor e somar cada linha da matriz
    {
        for(j=0;j<3;j++)
            linha = linha + mat[i][j]; //soma dos valores de cada linha
        printf(" \nLinha %d : %d \n ", i, linha);
        linha=0; //zerar a variável
    }

    printf(" Total por Coluna: \n\n");
    for(j=0;j<3;j++)//percorrer o vetor e somar cada coluna da matriz
    {
        for(i=0;i<3;i++)
            coluna = coluna + mat[i][j]; //soma dos valores de cada coluna
        printf(" \nColuna %d : %d \n ", j, coluna);
        coluna=0; //zerar a variável
    }

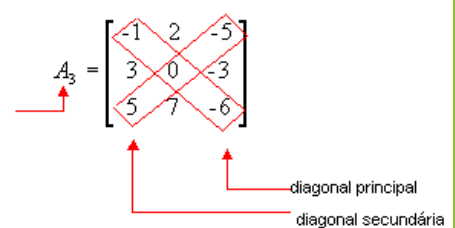
    system("pause");
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

► 2.2- Faça um programa que some a diagonal principal da matriz e apresente o seu resultado.

► 2.3 - Faça um programa que some a diagonal secundária da matriz e apresente o seu resultado.

OBS: ordem da matriz



Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

► 2.2-

```
#include <stdio.h>
#include <stdlib.h>

main( )
{
    //matriz 3x3
    int mat [3][3]={
        { 1,2,3},
        { 4,5,6},
        { 7,8,9}
    };

    int i, j;
    int soma=0;

    for( i = 0; i < 3; i++ )
    {
        for( j = 0; j < 3; j++ )
        {
            if( i==j )
            {
                soma += mat[i][j];
            }
        }
    }

    printf( "Soma da diagonal principal: %d", soma );
    system("pause");
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

► 2.3-

```
#include <stdio.h>
#include <stdlib.h>

main( )
{
    //matriz 3x3
    int mat [3][3]={
        { 1,2,3},
        { 4,5,6},
        { 7,8,9}
    };

    int i, j;
    int soma=0;

    for( i = 0; i < 3; i++ )
    {
        for( j = 0; j < 3; j++ )
        {
            if( j == (3 - 1 - i) )
            {
                soma += mat[i][j];
            }
        }
    }

    printf( "Soma da diagonal secundaria: %d", soma );
    system("pause");
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Exercícios Complementares

3- Faça um programa que lê uma matriz de 3 x 3 elementos usando um comando for, e que multiplique cada elemento por 5 e por fim, imprime o resultado.

4- Faça um programa que lê um vetor de 3 elementos e uma matriz de 3 x 3 elementos. Em seguida o programa deve fazer a multiplicação do vetor pelas colunas da matriz.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Exercícios Complementares

```
3- #include <stdio.h>
#include <stdlib.h>
main()
{
    int i, j, m[3][3];
    //captura os elementos
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
        {
            printf("Elemento[%d][%d]= ",i,j);
            scanf("%d",&m[i][j]);
        }
    //EXIBIR VALORES ORIGINAIS
    printf("\n::: Valores Originais ::: \n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }
    //multiplica por 5
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            m[i][j]=m[i][j]*5;
    //EXIBIR VALORES MULTIPLICADOS
    printf("\n::: Valores Multiplicados por 5::: \n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }
    system("pause");
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Exercícios Complementares (1/2)

```

4- #include <stdio.h>
    #include <stdlib.h>
    main()
    {
        int i, j, m[3][3], v[3];
        //captura os elementos do vetor
        printf("::: Informe os elementos do vetor :::\n");
        for(i=0;i<3;i++)
        {
            printf("Elemento[%d]= ",i);
            scanf("%d",&v[i]);
        }

        //captura os elementos da matriz
        printf("::: Informe os elementos da matriz :::\n");
        for(i=0;i<3;i++)
            for(j=0;j<3;j++)
            {
                printf("Elemento[%d][%d]= ",i,j);
                scanf("%d",&m[i][j]);
            }
    }

```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Exercícios Complementares (2/2)

```

4- //exibe valores originais
    printf("\n::: Valores Originais do Vetor :::\n");
    for(i=0;i<3;i++)
        printf("%d ",v[i]);

    printf("\n::: Valores Originais da Matriz :::\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }

    //multiplica vetor pelas linhas da matriz
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            m[i][j]=v[i]*m[i][j];

    //exibe valores multiplicados
    printf("\n::: Valores Multiplicados :::\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }

    system("pause");
}

```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Algoritmos e Práticas de Programação

Prof. Alessandro Santos

Funções

Funções

- O que são funções na linguagem C?

Funções

- ▶ Uma função nada mais é do que um bloco de código (ou seja, declarações e outros comandos) que podem ser nomeados e chamados de dentro de um programa. Em outras palavras, uma função é uma sequência de comandos que recebe um nome e pode ser chamada de qualquer parte do programa, quantas vezes forem necessárias, durante a execução do programa.
- ▶ A linguagem C possui muitas funções já implementadas que são utilizadas constantemente. Um exemplo delas são as funções básicas de entrada e saída: `scanf()` e `printf()`.
- ▶ O programador não precisa saber qual o código contido dentro das funções de entrada e saída para utilizá-las. Basta saber seu nome e como utilizá-la.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Por que utilizar funções?

- ▶ Duas são as principais razões para o uso de funções:
- ▶ Estruturação dos programas;
- ▶ Reutilização de código.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Modularização

- Modularização é uma técnica de programação que se caracteriza pela divisão de um programa em subprogramas.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Funções e Procedimentos em C

Procedimento - *Procedure*

Procedimento é um tipo de sub-rotina que se assemelha muito com um algoritmo e tem como objetivo executar uma ação que não retorna valor. Em C um procedimento é uma função do tipo **void**, ou seja, uma função sem retorno.

Função - *Function*

Uma sub-rotina do tipo função possui as mesmas características de uma procedure mas com uma importante diferença, a função **retorna** um valor ao término de sua execução.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Declaração de Funções

Na linguagem C, a declaração de uma função pelo programador segue a seguinte forma geral:

tipo retornado nome função (lista de parâmetros)

{

sequência de declarações e comandos

}

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Praticando... (Opção 1)

```
# include <stdio.h>
# include <stdlib.h>
//antes da função main
int Quadrado (int n) //recepção de parâmetro
{
    return (n*n);
}

int main()
{
    int n1,n2;
    printf("Digite um numero: \n");
    scanf("%d",&n1);

    n2= Quadrado(n1);// (passagem de parâmetro)
    .....
    printf("O quadrado do número digitado foi: %d \n", n2);

    system("pause");

    return(0);
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Praticando... (Opção 2)

```
# include <stdio.h>
# include <stdlib.h>
//Outra maneira de organizar funções
//Depois da função main
int Quadrado();//protótipo

int main()
{
    int n1,n2;
    printf("Digite um numero: \n");
    scanf("%d",&n1);

    n2= Quadrado(n1);// (passagem de parâmetro)

    printf("O quadrado do número digitado foi: %d \n", n2);

    system("pause");

    return(0);
}

int Quadrado (int n) //recepção de parâmetro
{
    return (n*n);
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Praticando... (mais de um parâmetro)

```
# include <stdio.h>
# include <stdlib.h>

int Maior (int x, int y) //recepção de parâmetros
{
    if(x>y)
        return x;
    else
        return y;
}

int main()
{
    int n1,n2,resultado;
    printf("Digite um numero: \n");
    scanf("%d",&n1);
    printf("Digite outro numero: \n");
    scanf("%d",&n2);

    resultado= Maior(n1,n2);// (passagem de parâmetro)

    printf("O numero maior digitado foi : %d \n", resultado);

    system("pause");

    return(0);
}
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Referências Bibliográficas

FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. *Lógica de Programação*. Ed. Pearson Brasil, 2000.

MANZANO, José Augusto Navarro Garcia; OLIVEIRA, Jair Figueiredo de. *Algoritmos: Lógica para o desenvolvimento de programação*. São Paulo: Érica, 2004.

MIZRAHI, Victorine Viviane. *Treinamento em linguagem C*. São Paulo: McGraw-Hill, 1990.

PEREIRA, Silvio do Lago. *Algoritmos e Lógica de Programação em C: Uma abordagem Didática*. São Paulo: Érica, 2010.

SANTOS, R. A. R. *Notas de Aula: Arquivo pessoal*. São Paulo: [s.e.], 2018.

SCHILD, Hebert. *C - Completo e total*. São Paulo: Makron Books, 1995.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Complemente o seu conhecimento

- Acesse o AVA e complemente o seu estudo com as aulas:

“Conceitos iniciais sobre Programação Estruturada em C”,
“Programação estruturada ou modular - tipos de módulos”,
“Parâmetros e passagem de parâmetros em programação estruturada”, e
“Passagem de vetor como parâmetro e macros”.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Desafio:

Faça os exercícios a seguir na linguagem de programação C. Lembre-se que para fazer o exercício a abstração é de suma importância!

- 1- Modularize o programa a seguir:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  main()
5  //início
6  int num1,num2,num3,num4;
7  //num1,num2,num3,num4: inteiro
8  int soma,subt,mult;
9  //soma, subt,mult: inteiro
10 float divi;
11 //divi: real
12
13
14 printf("Digite o primeiro número: ");
15 scanf("%d",&num1);
16 printf("Digite o segundo número: ");
17 scanf("%d",&num2);
18 printf("Digite o terceiro número: ");
19 scanf("%d",&num3);
20 printf("Digite o quarto número: ");
21 scanf("%d",&num4);
22
23 // Operações (processamento)
24 soma = num1 + num3;
25 subt = num2 - num4;
26 mult = num1 * num4;
27 divi = num3 / num4;
28
29 // Mostrar o resultado de cada operação (soma, subt, mult, divi)
30 printf("O resultado da soma é: %d\n", soma);
31 printf("O resultado da subtração é: %d\n", subt);
32 printf("O resultado da multiplicação é: %d\n", mult);
33 printf("O resultado da divisão é: %f\n", divi);
34
35 system("pause");
36 }

```

- 2- Melhore e modularize o programa incompleto a seguir:

```
switch (Origem)
{
case 1:
    printf ("\n %f Produto do Sul",Preco);
    break;
case 2:
    printf ("\n %f Produto do Norte",Preco);
    break;
case 3:
    printf ("\n %f Produto do Leste", Preco);
    break;
case 4:
    printf ("\n %f Produto do Oeste", Preco);
    break;
case 5:
    printf ("\n %f Produto do Sudeste", Preco);
    break;
case 6:
    printf ("\n %f Produto do Centro-Oeste", Preco);
    break;
case 7:
    printf ("\n %f Produto do Nordeste", Preco);
    break;
default:
    printf ("\n %f Produto importado", Preco);
    break;
}
```