

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

//Constantes
#define PI 3.14

float calculaVolume(float raio)
{
    return 4.0 / 3.0 * PI * pow(raio, 3);
}

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis
    float raio = 0;

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa calculo do volume da esfera>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    printf_s("\nDigite o raio: ");
    scanf_s("%f", &raio);

    printf("Volume da esfera: %f", calculaVolume(raio));

    //Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
    printf("\n\n\n");
    system("pause");

    //Retorno do método main
    return(EXIT_SUCCESS);
}
```

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

//Constantes
#define V0 "Nao e triangulo"
#define V1 "Triangulo Equilatero"
#define V2 "Triangulo Escaleno"
#define V3 "Triangulo Isosceles"
#define MENOR_VALOR_DE_LADO 0.1

int verificaTriangulo(float x, float y, float z)
{
    int retorno = 0;

    if ((x < y + z) && (z < x + y) && (y < z + x))
    {
        if (x == y && y == z)
        {
            retorno = 1;
        }
        else
        {
            if (x != y && y != z)
            {
                retorno = 2;
            }
            else
            {
                if ((x == y && y != z) || (y == z && z != x) || (z == x && x != y))
                {
                    retorno = 3;
                }
            }
        }
    }

    return retorno;
}

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis
    float x = 0;
    float y = 0;
    float z = 0;
    bool dadosDeEntradaValidos = true;

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa triangulos>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    do{
        dadosDeEntradaValidos = true;

        printf_s("\nDigite o lado x: ");
        scanf_s("%f", &x);

        //limpa o buffer do scanf
        fflush(stdin);

        if (x < MENOR_VALOR_DE_LADO)
        {
```

```
        printf("\n\nEntrada Invalida!\nO menor valor de um lado nao pode ser menor que %.2f.\n",
MENOR_VALOR_DE_LADO);

        dadosDeEntradaValidos = false;
    }
} while (!dadosDeEntradaValidos);

do{
    dadosDeEntradaValidos = true;

    printf_s("\nDigite o lado y: ");
    scanf_s("%f", &y);

    //limpa o buffer do scanf
    fflush(stdin);

    if (y < MENOR_VALOR_DE_LADO)
    {
        printf("\n\nEntrada Invalida!\nO menor valor de um lado nao pode ser menor que %.2f.\n",
MENOR_VALOR_DE_LADO);

        dadosDeEntradaValidos = false;
    }
} while (!dadosDeEntradaValidos);

do{
    dadosDeEntradaValidos = true;

    printf_s("\nDigite o lado z: ");
    scanf_s("%f", &z);

    //limpa o buffer do scanf
    fflush(stdin);

    if (z < MENOR_VALOR_DE_LADO)
    {
        printf("\n\nEntrada Invalida!\nO menor valor de um lado nao pode ser menor que %.2f.\n",
MENOR_VALOR_DE_LADO);

        dadosDeEntradaValidos = false;
    }
} while (!dadosDeEntradaValidos);

int resultado = verificaTriangulo(x, y, z);
char *texto = " ";

if (resultado == 1)
{
    texto = V1;
}
else
{
    if (resultado == 2)
    {
        texto = V2;
    }
    else
    {
        if (resultado == 3)
        {
            texto = V3;
        }
        else
        {
            if (resultado == 0)
            {

```

```
        texto = V0;
    }
}

printf_s("\n\nResposta da analise: %s", texto);

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

//Constantes
#define MEDIA_MENOR 0.0
#define MEDIA_MAIOR 10.0

char calculaConceito(float media)
{
    char conceito = ' ';

    if (media >= 0.0 && media < 5.0)
    {
        conceito = 'D';
    }
    else
    {
        if (media >= 5.0 && media < 7.0)
        {
            conceito = 'C';
        }
        else
        {
            if (media >= 7.0 && media < 9.0)
            {
                conceito = 'B';
            }
            else
            {
                if (media >= 9.0 && media <= 10.0)
                {
                    conceito = 'A';
                }
            }
        }
    }

    return conceito;
}

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis
    float media = 0;
    bool dadosDeEntradaValidos = true;

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa Media Final do Aluno>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    do{
        dadosDeEntradaValidos = true;

        printf_s("\nDigite a media final do aluno: ");
        scanf_s("%f", &media);

        //limpa o buffer do scanf
        fflush(stdin);

        if (media < MEDIA_MENOR || media > MEDIA_MAIOR)
        {
            printf("\n\nEntrada Invalida!\nA media nao pode ser menor que %.2f e maior que %.2f.\n", MEDIA_MENOR, MEDIA_MAIOR);
        }
    } while (!dadosDeEntradaValidos);
}
```

```
        dadosDeEntradaValidos = false;
    }
} while (!dadosDeEntradaValidos);

printf("Conceito: %c", calculaConceito(media));

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

//Constantes
#define LIMITE_MINIMO 0

int somatorio(int n)
{
    int somatorio = 0;

    for (int i = 1; i <= n; i++)
    {
        somatorio += i;
    }

    return somatorio;
}

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis
    int n = 0;
    bool dadosDeEntradaValidos = true;

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa somatorio de 0 a n>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    do{
        dadosDeEntradaValidos = true;

        printf_s("\nDigite o valor de n: ");
        scanf_s("%d", &n);

        //limpa o buffer do scanf
        fflush(stdin);

        if (n < LIMITE_MINIMO)
        {
            printf("\n\nEntrada Invalida!\nO valor de n nao pode ser menor que %d.\n", LIMITE_MINIMO);

            dadosDeEntradaValidos = false;
        }
    } while (!dadosDeEntradaValidos);

    printf("\n\nO somatorio de %d: %d", n, somatorio(n));

    //Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
    printf("\n\n\n");
    system("pause");

    //Retorno do método main
    return(EXIT_SUCCESS);
}
```

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

//Constantes
#define TAMANHO_DO_VETOR 100
#define MEDIA_MENOR 0.0
#define MEDIA_MAIOR 10.0

float mediaPonderada(float v[], int tamanho)
{
    float mediaPonderada = 0.0;
    int peso = 0;
    int somatorioDosPesos = 0;

    for (int i = 0; i < tamanho; i++)
    {
        peso = i + 1;
        mediaPonderada += v[i] * peso;
        somatorioDosPesos += peso;
    }

    mediaPonderada /= somatorioDosPesos;

    return mediaPonderada;
}

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis
    float vetor[TAMANHO_DO_VETOR];
    bool dadosDeEntradaValidos = true;

    for (int i = 0; i < TAMANHO_DO_VETOR; i++)
    {
        vetor[i] = 0.0;
    }

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa media ponderada em vetor por indice>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    for (int i = 0; i < TAMANHO_DO_VETOR; i++)
    {
        do{
            dadosDeEntradaValidos = true;

            printf_s("\nDigite o valor da posicao %d/%d do vetor: ", i + 1, TAMANHO_DO_VETOR);
            scanf_s("%f", &vetor[i]);

            //limpa o buffer do scanf
            fflush(stdin);

            if (vetor[i] < MEDIA_MENOR || vetor[i] > MEDIA_MAIOR)
            {
                printf("\n\nEntrada Invalida!\nA media nao pode ser menor que %.2f e maior que %.2f.\n",
                    MEDIA_MENOR, MEDIA_MAIOR);

                dadosDeEntradaValidos = false;
            }
        } while (!dadosDeEntradaValidos);
    }

    printf("\n\nMedia ponderada: %.2f", mediaPonderada(vetor, TAMANHO_DO_VETOR));
```



```
//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

//Constantes
#define QUANTIDADE_DE_LINHAS 9
#define QUANTIDADE_DE_COLUNAS 9

float somaElementosEmLinhasPares(float matriz[QUANTIDADE_DE_LINHAS][QUANTIDADE_DE_COLUNAS])
{
    float soma = 0.0;

    for (int i = 0; i < QUANTIDADE_DE_LINHAS; i++)
    {
        for (int j = 0; j < QUANTIDADE_DE_COLUNAS; j++)
        {
            if (i % 2 == 0)
            {
                soma += matriz[i][j];
            }
        }
    }

    return soma;
}

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis
    float m[QUANTIDADE_DE_LINHAS][QUANTIDADE_DE_COLUNAS];

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa soma dos elementos das linhas pares da matriz>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    for (int i = 0; i < QUANTIDADE_DE_LINHAS; i++)
    {
        for (int j = 0; j < QUANTIDADE_DE_COLUNAS; j++)
        {
            printf_s("\nDigite o valor da linha %d e coluna %d da matriz: ", i, j);
            scanf_s("%f", &m[i][j]);

            //limpa o buffer do scanf
            fflush(stdin);
        }
    }

    printf("\n\nSoma dos elementos das linhas pares: %.2f", somaElementosEmLinhasPares(m));

    //Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
    printf("\n\n\n");
    system("pause");

    //Retorno do método main
    return(EXIT_SUCCESS);
}
```

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <limits.h>

//Constantes
#define QUANTIDADE_DE_LINHAS 8
#define QUANTIDADE_DE_COLUNAS 8

float buscaMaiorElementoDaDiagonalPrincipal(float matriz[QUANTIDADE_DE_LINHAS][QUANTIDADE_DE_COLUNAS])
{
    float maiorValor = matriz[0][0];

    for (int i = 0; i < QUANTIDADE_DE_LINHAS; i++)
    {
        for (int j = 0; j < QUANTIDADE_DE_COLUNAS; j++)
        {
            if (i == 0 && j == 0)
            {
                ++j;
            }

            if (i == j)
            {
                if (maiorValor < matriz[i][j])
                {
                    maiorValor = matriz[i][j];
                }
            }
        }
    }

    return maiorValor;
}

void divideMatrizPorNumero(float matriz[QUANTIDADE_DE_LINHAS][QUANTIDADE_DE_COLUNAS], float numero)
{
    for (int i = 0; i < QUANTIDADE_DE_LINHAS; i++)
    {
        for (int j = 0; j < QUANTIDADE_DE_COLUNAS; j++)
        {
            matriz[i][j] /= numero;
        }
    }
}

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis
    float a[QUANTIDADE_DE_LINHAS][QUANTIDADE_DE_COLUNAS];

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa encontra maior elemento da diagonal principal matriz e divide  

    todos os elementos por esse valor>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    for (int i = 0; i < QUANTIDADE_DE_LINHAS; i++)
    {
        for (int j = 0; j < QUANTIDADE_DE_COLUNAS; j++)
        {
            printf_s("\nDigite o valor da linha %d e coluna %d da matriz: ", i, j);
            scanf_s("%f", &a[i][j]);

            //limpa o buffer do scanf
            fflush(stdin);
        }
    }
}
```

```
    }  
}  
  
printf("\n\nMatriz original\n\n");  
  
for (int i = 0; i < QUANTIDADE_DE_LINHAS; i++)  
{  
    for (int j = 0; j < QUANTIDADE_DE_COLUNAS; j++)  
    {  
        printf_s("%.2f ", a[i][j]);  
    }  
    printf_s("\n");  
}  
  
divideMatrizPorNumero(a, buscaMaiorElementoDaDiagonalPrincipal(a));  
  
printf("\n\nMatriz resultante\n\n");  
  
for (int i = 0; i < QUANTIDADE_DE_LINHAS; i++)  
{  
    for (int j = 0; j < QUANTIDADE_DE_COLUNAS; j++)  
    {  
        printf_s("%.2f ", a[i][j]);  
    }  
    printf_s("\n");  
}  
  
//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)  
printf("\n\n\n");  
system("pause");  
  
//Retorno do método main  
return(EXIT_SUCCESS);  
}
```