

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

//Constantes
#define MASCULINO 'M'
#define FEMININO 'F'
#define ALTURA_MINIMA 0.5
#define ALTURA_MAXIMA 2.6

//Método Main - Entry Point do Programa
int main()
{
    //Declaração e inicialização de Variáveis locais
    float altura = 0;
    char sexo = ' ';
    double pesoIdeal = 0.0;
    bool dadosDeEntradaValidos = true;

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

    //Captura de dados do teclado
    printf("\n\n\n =>> Digite seu sexo (M ou F): ");
    sexo = getchar();

    printf("\n =>> Digite sua altura em Metros (Minima: %.2f e Maxima: %.2f): ", ALTURA_MINIMA, ALTURA_MAXIMA);
    scanf_s("%f", &altura);

    //conversão para letra maiúscula
    sexo = toupper(sexo);

    //Validações
    if (sexo != MASCULINO && sexo != FEMININO)
    {
        dadosDeEntradaValidos = false;
    }

    if (altura < ALTURA_MINIMA || altura > ALTURA_MAXIMA)
    {
        dadosDeEntradaValidos = false;
    }

    //Cálculo do Peso Ideal
    if (dadosDeEntradaValidos)
    {
        if (sexo == MASCULINO)
        {
            pesoIdeal = (72.7 * altura) - 58;
        }
        else
        {
            if (sexo == FEMININO)
            {
                pesoIdeal = (62.1 * altura) - 44.7;
            }
        }

        //Mostra o resultado na tela
        printf("\n\n # O seu peso ideal e: %.2lf", pesoIdeal);
    }
    else
    {
        //Mostra uma mensagem de exceção
        printf("\n\n # Dado(s) de entrada invalido(s)!");
    }
}
```

```
//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>

//Constantes
#define SALARIO_MINIMO 724.0
#define PERCENTUAL_DE_AUMENTO_1 35.0
#define PERCENTUAL_DE_AUMENTO_2 15.0
#define VALOR_LIMITE_DO_PERCENTUAL_AUMENTO_1 1000.0
#define DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL 100.0

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis locais e iniciaização
    float salario = 0.0;
    double salarioComReajuste = 0.0;
    bool dadosDeEntradaValidos = true;

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

    //Captura de dados do teclado
    printf("\n\n\n => Digite seu salario (minimo: %.2f ): ", SALARIO_MINIMO);
    scanf_s("%f", &salario);

    //validacoes
    if (salario < SALARIO_MINIMO)
    {
        dadosDeEntradaValidos = false;
    }

    //Calculos de reajuste salarial
    if (dadosDeEntradaValidos)
    {
        salarioComReajuste = salario;

        if (salario <= VALOR_LIMITE_DO_PERCENTUAL_AUMENTO_1)
        {
            salarioComReajuste += salario * (PERCENTUAL_DE_AUMENTO_1 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);

            //Mostra o resultado na tela
            printf("\n\n # 0 seu salario com reajuste de %.0f%% e: %.2lf", PERCENTUAL_DE_AUMENTO_1,
salarioComReajuste);
        }
        else
        {
            salarioComReajuste += salario * (PERCENTUAL_DE_AUMENTO_2 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);

            //Mostra o resultado na tela
            printf("\n\n # 0 seu salario com reajuste de %.0f%% e: %.2lf", PERCENTUAL_DE_AUMENTO_2,
salarioComReajuste);
        }
    }
    else
    {
        //Mostra uma mensagem de exceção
        printf("\n\n # Dado(s) de entrada invalido(s)!");
    }

    //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>

//Constantes
#define SALDO_MINIMO_PARA_CONCESSAO_DE_CREDITO 0.0
#define DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL 100.0

enum percentualDeDesconto
{
    ACIMA_DE_4000_00 = 30,
    DE_3000_01_ATE_4000_00 = 25,
    DE_2000_01_ATE_3000_00 = 20,
    ATE_2000_00 = 10
};

typedef enum percentualDeDesconto tipoPercentualDeDesconto;

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis locais e iniciaização
    float saldoMedio = 0.0;
    double valorDoCreditoEspecialConcedido = 0.0;
    bool dadosDeEntradaValidos = true;
    tipoPercentualDeDesconto percentualDeDesconto = ACIMA_DE_4000_00;

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa Concessao de Credito Especial (Considerando o salario Medio)>>");
    printf("\n\n ## Autor: Leonardo Melo");

    //Captura de dados do teclado
    printf("\n\n\n => Digite seu saldo medio do ultimo ano: ");
    scanf_s("%f", &saldoMedio);

    //validacoes
    if (saldoMedio < SALDO_MINIMO_PARA_CONCESSAO_DE_CREDITO)
    {
        dadosDeEntradaValidos = false;
    }

    //Calculo da concessão de desconto
    if (dadosDeEntradaValidos)
    {
        if (saldoMedio >= SALDO_MINIMO_PARA_CONCESSAO_DE_CREDITO && saldoMedio <= 2000.0)
        {
            percentualDeDesconto = ATE_2000_00;
        }
        else
        {
            if (saldoMedio > 2000.0 && saldoMedio <= 3000.0)
            {
                percentualDeDesconto = DE_2000_01_ATE_3000_00;
            }
            else
            {
                if (saldoMedio > 3000.0 && saldoMedio <= 4000.0)
                {
                    percentualDeDesconto = DE_3000_01_ATE_4000_00;
                }
                else
                {
                    if (saldoMedio > 4000.0)
                    {
                        percentualDeDesconto = ACIMA_DE_4000_00;
                    }
                }
            }
        }
    }
}
```

```
}

switch (percentualDeDesconto)
{
case ACIMA_DE_4000_00:
    valorDoCreditoEspecialConcedido = saldoMedio * (((float)ACIMA_DE_4000_00) /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
    printf_s("\n\n # Valor do credito especial concedido considerando o percentual de %d%% e: %.2lf",
ACIMA_DE_4000_00, valorDoCreditoEspecialConcedido);
    break;
case DE_3000_01_ATE_4000_00:
    valorDoCreditoEspecialConcedido = saldoMedio * (((float)DE_3000_01_ATE_4000_00) /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
    printf_s("\n\n # Valor do credito especial concedido considerando o percentual de %d%% e: %.2lf",
DE_3000_01_ATE_4000_00, valorDoCreditoEspecialConcedido);
    break;
case DE_2000_01_ATE_3000_00:
    valorDoCreditoEspecialConcedido = saldoMedio * (((float)DE_2000_01_ATE_3000_00) /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
    printf_s("\n\n # Valor do credito especial concedido considerando o percentual de %d%% e: %.2lf",
DE_2000_01_ATE_3000_00, valorDoCreditoEspecialConcedido);
    break;
case ATE_2000_00:
    valorDoCreditoEspecialConcedido = saldoMedio * (((float)ATE_2000_00) /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
    printf_s("\n\n # Valor do credito especial concedido considerando o percentual de %d%% e: %.2lf",
ATE_2000_00, valorDoCreditoEspecialConcedido);
    break;
}
}
else
{
    //Mostra uma mensagem de exceção
    printf("\n\n # Dado(s) de entrada invalido(s)!");
}

//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>

//Constantes
#define VALOR_MINIMO_VEICULAR 1000
#define DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL 100.0

enum percentualDoDistribuidor
{
    PD_ACIMA_DE_40000_00 = 15,
    PD_DE_30000_01_ATE_40000_00 = 10,
    PD_ATE_30000_00 = 5
};

typedef enum percentualDoDistribuidor tipoPercentualDoDistribuidor;

enum percentualDoImposto
{
    PI_ACIMA_DE_40000_00 = 20,
    PI_DE_30000_01_ATE_40000_00 = 15,
    PI_ATE_30000_00 = 0
};

typedef enum percentualDoImposto tipoPercentualDoImposto;

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis locais
    float custoDeFabrica = 0.0;
    double custoAoConsumidor = 0.0;
    double valorPagoAoDistribuidor = 0.0;
    double valorDoImposto = 0.0;
    tipoPercentualDoDistribuidor percentualDoDistribuidor = PD_ACIMA_DE_40000_00;
    tipoPercentualDoImposto percentualDoImposto = PI_ACIMA_DE_40000_00;
    bool dadosDeEntradaValidos = true;

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

    //Captura de dados do teclado
    printf("\n\n\n >> Digite o custo de fabrica do veiculo: ");
    scanf_s("%f", &custoDeFabrica);

    //validacoes
    if (custoDeFabrica < VALOR_MINIMO_VEICULAR)
    {
        dadosDeEntradaValidos = false;
    }

    //Calculo da concessão de desconto
    if (dadosDeEntradaValidos)
    {
        if (custoDeFabrica >= VALOR_MINIMO_VEICULAR && custoDeFabrica <= 30000.0)
        {
            percentualDoDistribuidor = PD_ATE_30000_00;
            percentualDoImposto = PI_ATE_30000_00;
        }
        else
        {
            if (custoDeFabrica > 30000.0 && custoDeFabrica <= 40000.0)
            {
                percentualDoDistribuidor = PD_DE_30000_01_ATE_40000_00;
                percentualDoImposto = PI_DE_30000_01_ATE_40000_00;
            }
        }
    }
}
```

```

    else
    {
        if (custoDeFabrica > 40000.0)
        {
            percentualDoDistribuidor = PD_ACIMA_DE_40000_00;
            percentualDoImposto = PI_ACIMA_DE_40000_00;
        }
    }

    switch (percentualDoDistribuidor)
    {
        case PD_ACIMA_DE_40000_00:
            valorPagoAoDistribuidor = custoDeFabrica * ((float)PD_ACIMA_DE_40000_00 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            valorDoImposto = custoDeFabrica * ((float)PI_ACIMA_DE_40000_00 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            custoAoConsumidor = custoDeFabrica + valorPagoAoDistribuidor + valorDoImposto;
            printf_s("\n\n # Custo ao consumidor (Distribuidor = %d%% e Imposto = %d%%): %.21f",
PD_ACIMA_DE_40000_00, PI_ACIMA_DE_40000_00, custoAoConsumidor);
            break;
        case PD_DE_30000_01_ATE_40000_00:
            valorPagoAoDistribuidor = custoDeFabrica * ((float)PD_DE_30000_01_ATE_40000_00 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            valorDoImposto = custoDeFabrica * ((float)PI_DE_30000_01_ATE_40000_00 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            custoAoConsumidor = custoDeFabrica + valorPagoAoDistribuidor + valorDoImposto;
            printf_s("\n\n # Custo ao consumidor (Distribuidor = %d%% e Imposto = %d%%): %.21f",
PD_DE_30000_01_ATE_40000_00, PI_DE_30000_01_ATE_40000_00, custoAoConsumidor);
            break;
        case PD_ATE_30000_00:
            valorPagoAoDistribuidor = custoDeFabrica * ((float)PD_ATE_30000_00 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            valorDoImposto = custoDeFabrica * ((float)PI_ATE_30000_00 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            custoAoConsumidor = custoDeFabrica + valorPagoAoDistribuidor + valorDoImposto;
            printf_s("\n\n # Custo ao consumidor (Distribuidor = %d%% e Imposto = %d%%): %.21f", PD_ATE_30000_00
, PI_ATE_30000_00, custoAoConsumidor);
            break;
    }
}
else
{
    //Mostra uma mensagem de exceção
    printf("\n\n # Dado(s) de entrada invalido(s)!");
}

//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 <ctype.h>

//Constantes
#define HORA_CODIGO_A 15.0
#define HORA_CODIGO_B 18.5
#define HORA_CODIGO_C 21.0

#define CODIGO_CATEGORIA_A 'A'
#define CODIGO_CATEGORIA_B 'B'
#define CODIGO_CATEGORIA_C 'C'

#define PERIODO_DE_PAGAMENTO_EM_SEMANAS 4.5

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis locais
    float numeroDeHorasTrabalhadasSemanaisPeloProfessor = 0.0;
    char codigoDoProfessor = 'A';
    double valorASerPagoParaProfessorEm4SemanaisEMeia = 0.0;
    bool dadosDeEntradaValidos = true;

    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa Salario do Professor em 4 semanas e meia>>");
    printf("\n\n ## Autor: Leonardo Melo");

    //Captura de dados do teclado
    printf("\n\n\n => Digite o codigo da categoria do profesor (A, B ou C): ");
    codigoDoProfessor = getchar();

    //Captura de dados do teclado
    printf("\n\n\n => Digite o numero de horas semanais do professor: ");
    scanf_s("%f", &numeroDeHorasTrabalhadasSemanaisPeloProfessor);

    //conversão para letra maiúscula
    codigoDoProfessor = toupper(codigoDoProfessor);

    //validacoes
    if (codigoDoProfessor != CODIGO_CATEGORIA_A && codigoDoProfessor != CODIGO_CATEGORIA_B && codigoDoProfessor != CODIGO_CATEGORIA_C)
    {
        dadosDeEntradaValidos = false;
    }

    //Calculo e Exibição de resposta
    if (dadosDeEntradaValidos)
    {
        if (codigoDoProfessor == CODIGO_CATEGORIA_A)
        {
            valorASerPagoParaProfessorEm4SemanaisEMeia = numeroDeHorasTrabalhadasSemanaisPeloProfessor *
            PERIODO_DE_PAGAMENTO_EM_SEMANAS * HORA_CODIGO_A;
        }
        else
        {
            if (codigoDoProfessor == CODIGO_CATEGORIA_B)
            {
                valorASerPagoParaProfessorEm4SemanaisEMeia = numeroDeHorasTrabalhadasSemanaisPeloProfessor *
                PERIODO_DE_PAGAMENTO_EM_SEMANAS * HORA_CODIGO_B;
            }
            else
            {
                if (codigoDoProfessor == CODIGO_CATEGORIA_C)
                {
                    valorASerPagoParaProfessorEm4SemanaisEMeia = numeroDeHorasTrabalhadasSemanaisPeloProfessor *

```

```
        PERIODO_DE_PAGAMENTO_EM_SEMANAS * HORA_CODIGO_C;
    }
}

    printf("\n\n # Valor a receber referente ao periodo de 4 semanas e meia: %.2f",
valorASerPagoParaProfessorEm4SemanaisEMeia);
}
else
{
    //Mostra uma mensagem de exceção
    printf("\n\n # Dado(s) de entrada invalido(s)!");
}

//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>

//Constantes
#define TEXTO_CATEGORIA_DE_PRODUTO_1 "Alimento Nao-Perecivel"
#define TEXTO_CATEGORIA_DE_PRODUTO_2_3_4 "Alimento Perecivel"
#define TEXTO_CATEGORIA_DE_PRODUTO_5_6 "Vetuario"
#define TEXTO_CATEGORIA_DE_PRODUTO_7 "Higiene pessoal"
#define TEXTO_CATEGORIA_DE_PRODUTO_8_A_15 "Limpeza e Utensilios Domesticos"

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

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

    //Captura de dados do teclado
    printf("\n\n\n =>> Digite o codigo do produto: ");
    scanf_s("%d", &codigoDoProduto);

    //validacoes
    if (codigoDoProduto < 1 || codigoDoProduto > 15)
    {
        dadosDeEntradaValidos = false;
    }

    //Calculo e Exibição de resposta
    if (dadosDeEntradaValidos)
    {
        switch (codigoDoProduto)
        {
            case 1:
                printf("\n\n # Codigo do Produto: %d - Categoria do Produto: %s", codigoDoProduto,
TEXTO_CATEGORIA_DE_PRODUTO_1);
                break;
            case 2: case 3: case 4:
                printf("\n\n # Codigo do Produto: %d - Categoria do Produto: %s", codigoDoProduto,
TEXTO_CATEGORIA_DE_PRODUTO_2_3_4);
                break;
            case 5: case 6:
                printf("\n\n # Codigo do Produto: %d - Categoria do Produto: %s", codigoDoProduto,
TEXTO_CATEGORIA_DE_PRODUTO_5_6);
                break;
            case 7:
                printf("\n\n # Codigo do Produto: %d - Categoria do Produto: %s", codigoDoProduto,
TEXTO_CATEGORIA_DE_PRODUTO_7);
                break;
            default:
                printf("\n\n # Codigo do Produto: %d - Categoria do Produto: %s", codigoDoProduto,
TEXTO_CATEGORIA_DE_PRODUTO_8_A_15);
                break;
        }
    }
    else
    {
        //Mostra uma mensagem de exceção
        printf("\n\n # Dado(s) de entrada invalido(s)!");
    }

    //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>

//Constantes
#define TEXTO_CATEGORIA_DO_NADADOR_IDADE_5_A_7 "Infantil"
#define TEXTO_CATEGORIA_DO_NADADOR_IDADE_8_A_10 "Juvenil"
#define TEXTO_CATEGORIA_DO_NADADOR_IDADE_11_A_15 "Adolescente"
#define TEXTO_CATEGORIA_DO_NADADOR_IDADE_16_A_30 "Adulto"
#define TEXTO_CATEGORIA_DO_NADADOR_IDADE_ACIMA_DE_30 "Senior"

#define IDADE_MINIMA_NADADOR 5
#define IDADE_MAXIMA_NADADOR 65

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

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

    //Captura de dados do teclado
    printf("\n\n\n =>> Digite a idade do nadador: ");
    scanf_s("%d", &idadeDoNadador);

    //validacoes
    if (idadeDoNadador < IDADE_MINIMA_NADADOR || idadeDoNadador > IDADE_MAXIMA_NADADOR)
    {
        dadosDeEntradaValidos = false;
    }

    //Calculo e Exibição de resposta
    if (dadosDeEntradaValidos)
    {
        if (idadeDoNadador >= 5 && idadeDoNadador <= 7)
        {
            printf("\n\n # Idade do Nadador: %d - Categoria do Nadador: %s", idadeDoNadador,
TEXTO_CATEGORIA_DO_NADADOR_IDADE_5_A_7);
        }
        else
        {
            if (idadeDoNadador >= 8 && idadeDoNadador <= 10)
            {
                printf("\n\n # Idade do Nadador: %d - Categoria do Nadador: %s", idadeDoNadador,
TEXTO_CATEGORIA_DO_NADADOR_IDADE_8_A_10);
            }
            else
            {
                if (idadeDoNadador >= 11 && idadeDoNadador <= 15)
                {
                    printf("\n\n # Idade do Nadador: %d - Categoria do Nadador: %s", idadeDoNadador,
TEXTO_CATEGORIA_DO_NADADOR_IDADE_11_A_15);
                }
                else
                {
                    if (idadeDoNadador >= 16 && idadeDoNadador <= 30)
                    {
                        printf("\n\n # Idade do Nadador: %d - Categoria do Nadador: %s", idadeDoNadador,
TEXTO_CATEGORIA_DO_NADADOR_IDADE_16_A_30);
                    }
                    else
                    {
                        printf("\n\n # Idade do Nadador: %d - Categoria do Nadador: %s", idadeDoNadador,
TEXTO_CATEGORIA_DO_NADADOR_IDADE_ACIMA_DE_30);
                    }
                }
            }
        }
    }
}
```

```
        }
    }
}

}
else
{
    //Mostra uma mensagem de exceção
    printf("\n\n # Dado(s) de entrada invalido(s)!");
}

//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>

//Constantes
#define PERCENTUAL_IMPOSTO_ESTADO_1 35.0
#define PERCENTUAL_IMPOSTO_ESTADO_2 25.0
#define PERCENTUAL_IMPOSTO_ESTADO_3 15.0
#define PERCENTUAL_IMPOSTO_ESTADO_4 5.0
#define PERCENTUAL_IMPOSTO_ESTADO_5 0.0

#define PRECO_DO_QUILO_PARA_CODIGO_DE_CARGA_DE_10_A_20 100.0
#define PRECO_DO_QUILO_PARA_CODIGO_DE_CARGA_DE_21_A_30 250.0
#define PRECO_DO_QUILO_PARA_CODIGO_DE_CARGA_DE_31_A_40 340.0

#define CODIGO_MAXIMO_DE_ESTADO 5
#define CODIGO_MINIMO_DE_ESTADO 1

#define CODIGO_MAXIMO_DE_CARGA 40
#define CODIGO_MINIMO_DE_CARGA 10

#define PESO_MINIMO_DA_CARGA_DO_CAMINHAO 0

#define MULTIPLICADOR_PARA_TRANSFORMAR_TONELADAS_EM_QUILOS 1000

#define DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL 100.0

//Método Main - Entry Point do Programa
int main()
{
    //Declaração de variáveis locais
    int codigoEstadoDaCarga = 0;
    int codigoDaCarga = 0;
    float pesoDaCargaEmToneladas = 0.0;
    float pesoDaCargaEmQuilos = 0.0;
    float precoDaCarga = 0.0;
    double valorDoImposto = 0.0;
    double precoTotalDaCarga = 0.0;
    bool dadosDeEntradaValidos = true;

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

    //Captura de dados do teclado
    printf("\n\n\n =>> Digite o codigo do estado de origem da carga do caminhao: ");
    scanf_s("%d", &codigoEstadoDaCarga);

    //Captura de dados do teclado
    printf("\n\n\n =>> Digite o digite o peso da carga do caminhão (toneladas): ");
    scanf_s("%f", &pesoDaCargaEmToneladas);

    //Captura de dados do teclado
    printf("\n\n\n =>> Digite o digite o codigo da carga do caminhao: ");
    scanf_s("%d", &codigoDaCarga);

    //validacoes
    if (codigoEstadoDaCarga < CODIGO_MINIMO_DE_ESTADO || codigoEstadoDaCarga > CODIGO_MAXIMO_DE_ESTADO)
    {
        dadosDeEntradaValidos = false;
    }

    //validacoes
    if (pesoDaCargaEmToneladas < PESO_MINIMO_DA_CARGA_DO_CAMINHAO)
    {
        dadosDeEntradaValidos = false;
    }
}
```

```
//validacoes
if (codigoDaCarga < CODIGO_MINIMO_DE_CARGA || codigoDaCarga > CODIGO_MAXIMO_DE_CARGA)
{
    dadosDeEntradaValidos = false;
}

//Calculo e Exibição de resposta
if (dadosDeEntradaValidos)
{
    pesoDaCargaEmQuilos = pesoDaCargaEmToneladas * MULTIPLICADOR_PARA_TRANSFORMAR_TONELADAS_EM_QUILOS;

    printf("\n\n # Peso da carga em KG: %.2f", pesoDaCargaEmQuilos);

    if (codigoDaCarga >= 10 && codigoDaCarga <= 20)
    {
        precoDaCarga = pesoDaCargaEmQuilos * (float)PRECO_DO_QUILO_PARA_CODIGO_DE_CARGA_DE_10_A_20;
    }
    else
    {
        if (codigoDaCarga >= 21 && codigoDaCarga <= 30)
        {
            precoDaCarga = pesoDaCargaEmQuilos * (float)PRECO_DO_QUILO_PARA_CODIGO_DE_CARGA_DE_21_A_30;
        }
        else
        {
            precoDaCarga = pesoDaCargaEmQuilos * (float)PRECO_DO_QUILO_PARA_CODIGO_DE_CARGA_DE_31_A_40;
        }
    }

    printf("\n\n # Preco da carga: R$ %.2f", precoDaCarga);

    switch (codigoEstadoDaCarga)
    {
        case 1:
            valorDoImposto = precoDaCarga * (PERCENTUAL_IMPOSTO_ESTADO_1 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            break;
        case 2:
            valorDoImposto = precoDaCarga * (PERCENTUAL_IMPOSTO_ESTADO_2 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            break;
        case 3:
            valorDoImposto = precoDaCarga * (PERCENTUAL_IMPOSTO_ESTADO_3 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            break;
        case 4:
            valorDoImposto = precoDaCarga * (PERCENTUAL_IMPOSTO_ESTADO_4 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            break;
        case 5:
            valorDoImposto = precoDaCarga * (PERCENTUAL_IMPOSTO_ESTADO_5 /
DENOMINADOR_DE_TRANSFORMACAO_PERCENTUAL_PARA_DECIMAL);
            break;
    }

    printf("\n\n # Valor do imposto: R$ %.21f", valorDoImposto);

    precoTotalDaCarga = precoDaCarga + valorDoImposto;

    printf("\n\n # Preco total da Carga: R$ %.21f", precoTotalDaCarga);
}
else
{
    //Mostra uma mensagem de exceção
    printf("\n\n # Dado(s) de entrada invalido(s)!");
}

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
```



```
printf("\n\n\n");  
system("pause");  
  
//Retorno do método main  
return(EXIT_SUCCESS);  
}
```