

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

#define TAMANHO_DO_VETOR 10
#define QUANTIDADE_MINIMA_DE_PECAS_VENDIDAS 1
#define PRECO_MINIMO_DE_UMA_PECA_A_SER_VENDIDA 0.01

//Método Main - Entry Point do Programa
int main()
{
    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa Vendedores>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    //Declaração de variáveis locais
    int quantidadeDePecasVendidas[TAMANHO_DO_VETOR];
    double precoDaPecaVendida[TAMANHO_DO_VETOR];
    double valorTotalDaVenda[TAMANHO_DO_VETOR];
    bool dadosDeEntradaValidos;
    int quantidadeTotalDePecasVendidasPorTodosVendedores = 0;

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

            printf("\nInforme a quantidade de pecas vendidas pelo vendedor %d: ", i + 1);
            scanf_s("%d", &quantidadeDePecasVendidas[i]);

            if (quantidadeDePecasVendidas[i] < QUANTIDADE_MINIMA_DE_PECAS_VENDIDAS)
            {
                printf("\n\nEntrada Invalida!\nA quantidade de pecas vendidas nao pode ser menor que %d.\n",
QUANTIDADE_MINIMA_DE_PECAS_VENDIDAS);
                dadosDeEntradaValidos = false;
            }
        } while (!dadosDeEntradaValidos);

        do{
            dadosDeEntradaValidos = true;

            printf("\nInforme o preco da peca vendida pelo vendedor %d: ", i + 1);
            scanf_s("%lf", &precoDaPecaVendida[i]);

            if (precoDaPecaVendida[i] < PRECO_MINIMO_DE_UMA_PECA_A_SER_VENDIDA)
            {
                printf("\n\nEntrada Invalida!\nO preco de uma peca a ser vendida nao pode ser menor que %.2lf.\n",
PRECO_MINIMO_DE_UMA_PECA_A_SER_VENDIDA);
                dadosDeEntradaValidos = false;
            }
        } while (!dadosDeEntradaValidos);

        quantidadeTotalDePecasVendidasPorTodosVendedores += quantidadeDePecasVendidas[i];

        valorTotalDaVenda[i] = quantidadeDePecasVendidas[i] * precoDaPecaVendida[i];
    }

    printf_s("\n\n\nQuantidade total de pecas vendidas por todos vendedores: %d\n",
quantidadeTotalDePecasVendidasPorTodosVendedores);

    for (int i = 0; i < TAMANHO_DO_VETOR; i++)
    {
        printf_s("\nVendedor %4d, valor total da sua venda: R$ %.2lf ", i + 1, valorTotalDaVenda[i]);
```

```
}

//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>

#define QUANTIDADE_DE_QUESTOES 10
#define QUANTIDADE_DE_ALUNOS 15
#define VALOR_POR_QUESTAO_CORRETA 1.0
#define NOTA_MINIMA_PARA_APROVACAO 7.0
#define MENOR_NUMERO_DE_IDENTIFICACAO_DE_ALUNO 1

#define RESPOSTA_A 'A'
#define RESPOSTA_B 'B'
#define RESPOSTA_C 'C'
#define RESPOSTA_D 'D'
#define RESPOSTA_E 'E'

//Método Main - Entry Point do Programa
int main()
{
    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa Correcao de Provas>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    //Declaração de variáveis locais
    char gabarito[QUANTIDADE_DE_QUESTOES];

    int numeroDosAlunos[QUANTIDADE_DE_ALUNOS];

    char provasDosAlunos[QUANTIDADE_DE_ALUNOS][QUANTIDADE_DE_QUESTOES];

    bool dadosDeEntradaValidos;

    int quantidadeDeAlunosAprovados = 0;

    //Ler gabarito
    for (int numeroDaQuestao = 0; numeroDaQuestao < QUANTIDADE_DE_QUESTOES; numeroDaQuestao++)
    {
        do{
            dadosDeEntradaValidos = true;

            printf("\nInforme o gabarito da questao <<< %2d >>> (Opcoes => %c, %c, %c, %c ou %c): ",
                numeroDaQuestao + 1, RESPOSTA_A, RESPOSTA_B, RESPOSTA_C, RESPOSTA_D, RESPOSTA_E);
            gabarito[numeroDaQuestao] = getchar();

            fflush(stdin);

            gabarito[numeroDaQuestao] = toupper(gabarito[numeroDaQuestao]);

            if (gabarito[numeroDaQuestao] != RESPOSTA_A && gabarito[numeroDaQuestao] != RESPOSTA_B && gabarito
                [numeroDaQuestao] != RESPOSTA_C && gabarito[numeroDaQuestao] != RESPOSTA_D && gabarito[numeroDaQuestao] !=
                RESPOSTA_E)
            {
                printf("\n\nEntrada Invalida!\nO gabarito nao pode ser diferente de %c, %c, %c, %c e %c.\n",
                    RESPOSTA_A, RESPOSTA_B, RESPOSTA_C, RESPOSTA_D, RESPOSTA_E);

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

    //Ler respostas dos alunos
    for (int numeroDoAluno = 0; numeroDoAluno < QUANTIDADE_DE_ALUNOS; numeroDoAluno++)
    {
        //Ler Numero do aluno
        do{
```

```
    dadosDeEntradaValidos = true;

    printf("\nInforme o numero de identificacao do aluno <<< %2d >>>: ", numeroDoAluno + 1);

    scanf_s("%d", &numeroDosAlunos[numeroDoAluno]);

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

    if (numeroDosAlunos[numeroDoAluno] < MENOR_NUMERO_DE_IDENTIFICACAO_DE_ALUNO)
    {
        printf("\n\nEntrada Invalida!\nO numero nao pode ser menor que %d.\n",
        MENOR_NUMERO_DE_IDENTIFICACAO_DE_ALUNO);
        dadosDeEntradaValidos = false;
    }

    } while (!dadosDeEntradaValidos);

    for (int numeroDaQuestao = 0; numeroDaQuestao < QUANTIDADE_DE_QUESTOES; numeroDaQuestao++)
    {
        do{
            dadosDeEntradaValidos = true;

            printf("\nInforme a resposta da questao <<< %2d >>> do aluno <<< %2d >>>: ", numeroDaQuestao + 1
, numeroDoAluno + 1);

            provasDosAlunos[numeroDoAluno][numeroDaQuestao] = getchar();

            fflush(stdin);

            provasDosAlunos[numeroDoAluno][numeroDaQuestao] = toupper(provasDosAlunos[numeroDoAluno]
[numeroDaQuestao]);

            if (provasDosAlunos[numeroDoAluno][numeroDaQuestao] != RESPOSTA_A && provasDosAlunos
[numeroDoAluno][numeroDaQuestao] != RESPOSTA_B && provasDosAlunos[numeroDoAluno][numeroDaQuestao] !=
RESPOSTA_C && provasDosAlunos[numeroDoAluno][numeroDaQuestao] != RESPOSTA_D && provasDosAlunos
[numeroDoAluno][numeroDaQuestao] != RESPOSTA_E)
            {
                printf("\n\nEntrada Invalida!\nA questao nao pode ser diferente de %c, %c, %c, %c e %c.\n",
RESPOSTA_A, RESPOSTA_B, RESPOSTA_C, RESPOSTA_D, RESPOSTA_E);
                dadosDeEntradaValidos = false;
            }

        } while (!dadosDeEntradaValidos);

    }

    printf("\n");
}

for (int numeroDoAluno = 0; numeroDoAluno < QUANTIDADE_DE_ALUNOS; numeroDoAluno++)
{
    printf_s("\n\nAluno <<%d>>\n", numeroDoAluno + 1);
    printf_s("Numero de identificacao: %d\n", numeroDosAlunos[numeroDoAluno]);

    float nota = 0.0;

    for (int numeroDaQuestao = 0; numeroDaQuestao < QUANTIDADE_DE_QUESTOES; numeroDaQuestao++)
    {
        if (provasDosAlunos[numeroDoAluno][numeroDaQuestao] == gabarito[numeroDaQuestao])
        {
            nota += VALOR_POR_QUESTAO_CORRETA;
        }
    }

    if (nota >= NOTA_MINIMA_PARA_APROVACAO)
```

```
{
    quantidadeDeAlunosAprovados++;
}

printf_s("Nota: %.2f", nota);
}

printf_s("\n\nPercentual de alunos aprovados: %.2lf%%", (100.0 * (double)quantidadeDeAlunosAprovados) /
(double)QUANTIDADE_DE_ALUNOS);

//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>

#define QUANTIDADE_DE_MES 12
#define ZERO_ABSOLUTO -273.15
#define MAIOR_TEMPERATURA_REGISTRADA_NA_TERRA 70.0

//Método Main - Entry Point do Programa
int main()
{
    char* meses[QUANTIDADE_DE_MES] = { "JANEIRO", "FEVEREIRO", "MARCO", "ABRIL", "MAIO", "JUNHO", "JULHO",
    "AGOSTO", "SETEMBRO", "OUTUBRO", "NOVEMBRO", "DEZEMBRO" };

    float temperaturaCelsiusMediaDosMeses[QUANTIDADE_DE_MES];

    bool dadosDeEntradaValidos;

    float maiorTemperatura = ZERO_ABSOLUTO;
    float menorTemperatura = MAIOR_TEMPERATURA_REGISTRADA_NA_TERRA;

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

    for (int mes = 0; mes < QUANTIDADE_DE_MES; mes++)
    {
        //Ler Numero do aluno
        do{
            dadosDeEntradaValidos = true;

            printf("\nInforme a temperatura media (Celsius) do mes de %s: ", meses[mes]);

            scanf_s("%f", &temperaturaCelsiusMediaDosMeses[mes]);

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

            if (temperaturaCelsiusMediaDosMeses[mes] < ZERO_ABSOLUTO || temperaturaCelsiusMediaDosMeses[mes] >
            MAIOR_TEMPERATURA_REGISTRADA_NA_TERRA)
            {
                printf("\n\nEntrada Invalida!\nA temperatura media (Celsius) nao pode ser menor que %.2f (Zero
                Absoluto) ou maior que %.2f.\n", ZERO_ABSOLUTO, MAIOR_TEMPERATURA_REGISTRADA_NA_TERRA);

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

        if (temperaturaCelsiusMediaDosMeses[mes] > maiorTemperatura)
        {
            maiorTemperatura = temperaturaCelsiusMediaDosMeses[mes];
        }

        if (temperaturaCelsiusMediaDosMeses[mes] < menorTemperatura)
        {
            menorTemperatura = temperaturaCelsiusMediaDosMeses[mes];
        }
    }

    printf_s("\n\nMenor temperatura media: %.2f\n", menorTemperatura);
    printf_s("Mese(s) com menor tempetatura media: \n");

    for (int mes = 0; mes < QUANTIDADE_DE_MES; mes++)
    {
        if (temperaturaCelsiusMediaDosMeses[mes] == menorTemperatura)
```

```
        {
            printf_s("\n%d - %s", mes + 1, meses[mes]);
        }
    }

    printf_s("\n\nMaior temperatura media: %.2f\n", maiorTemperatura);
    printf_s("Mese(s) com maior tempetatura media: \n");

    for (int mes = 0; mes < QUANTIDADE_DE_MES; mes++)
    {
        if (temperaturaCelsiusMediaDosMeses[mes] == maiorTemperatura)
        {
            printf_s("\n%d - %s", mes + 1, meses[mes]);
        }
    }

    //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>

#define TAMANHO_VETOR_1 10
#define TAMANHO_VETOR_2 5
#define TAMANHO_VETOR_RESULTANTE_1 10
#define TAMANHO_VETOR_RESULTANTE_2 10

#define MENOR_NUMERO_DO_TIPO_INTEIRO -858993460

//Método Main - Entry Point do Programa
int main()
{
    int vetor1[TAMANHO_VETOR_1];
    int vetor2[TAMANHO_VETOR_2];

    int vetorResultante1[TAMANHO_VETOR_RESULTANTE_1];
    int vetorResultante2[TAMANHO_VETOR_RESULTANTE_2];

    int somaDosElementosDoVetor2 = 0;

    bool dadosDeEntradaValidos;

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

    for (int i = 0; i < TAMANHO_VETOR_1; i++)
    {
        do{
            dadosDeEntradaValidos = true;

            printf("\nInforme um numero inteiro para o vetor (1) na posicao (%d): ", i);

            scanf_s("%d", &vetor1[i]);

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

            if (vetor1[i] == MENOR_NUMERO_DO_TIPO_INTEIRO)
            {
                printf("\n\nEntrada Invalida!\nDigite somente numeros.\n\n");

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

    printf_s("\n\n");

    for (int i = 0; i < TAMANHO_VETOR_2; i++)
    {
        do{
            dadosDeEntradaValidos = true;

            printf("\nInforme um numero inteiro para o vetor (2) na posicao (%d): ", i);

            scanf_s("%d", &vetor2[i]);

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

            if (vetor2[i] == MENOR_NUMERO_DO_TIPO_INTEIRO)
            {
                printf("\n\nEntrada Invalida!\nDigite somente numeros.\n\n");
            }
        } while (!dadosDeEntradaValidos);
    }
}
```



```
        dadosDeEntradaValidos = false;
    }

    } while (!dadosDeEntradaValidos);

    somaDosElementosDoVetor2 += vetor2[i];
}

int contadorVetorResultante1 = 0;
int contadorVetorResultante2 = 0;

for (int i = 0; i < TAMANHO_VETOR_1; i++)
{
    if (vetor1[i] % 2 == 0) //elemento de vetor1 é par
    {
        vetorResultante1[contadorVetorResultante1] = vetor1[i] + somaDosElementosDoVetor2;
        contadorVetorResultante1++;
    }
    else //elemento de vetor1 é ímpar
    {
        vetorResultante2[contadorVetorResultante2] = vetor1[i] + somaDosElementosDoVetor2;
        contadorVetorResultante2++;
    }
}

printf_s("\n\nElementos do VetorResultante1:\n\n");

for (int i = 0; i < TAMANHO_VETOR_RESULTANTE_1; i++)
{
    if (vetorResultante1[i] != MENOR_NUMERO_DO_TIPO_INTEIRO)
    {
        printf_s("%d\n", vetorResultante1[i]);
    }
    else
    {
        printf_s("POSICAO DO VETOR VAZIA\n");
    }
}

printf_s("\n\nElementos do VetorResultante2:\n\n");

for (int i = 0; i < TAMANHO_VETOR_RESULTANTE_2; i++)
{
    if (vetorResultante2[i] != MENOR_NUMERO_DO_TIPO_INTEIRO)
    {
        printf_s("%d\n", vetorResultante2[i]);
    }
    else
    {
        printf_s("POSICAO DO VETOR VAZIA\n");
    }
}

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

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