



Estácio

1). Faça um programa que solicite o número de elementos de vetor, solicite os elementos e armazene-os no vetor, e imprima a quantidade de elementos pares e ímpares

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

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

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

```
int main() {
```

```
    int n, i;
```

```
    printf("Digite o numero de elementos do vetor: ");
```

```
    scanf("%d", &n);
```

```
    int vetor[n];
```

```
    printf("Digite os elementos do vetor:\n");
```

```
    for (i = 0; i < n; i++) {
```

```
        scanf("%d", &vetor[i]); }
```

```
    int pares = 0, impares = 0;
```

```
    for (i = 0; i < n; i++) {
```

```
        if (vetor[i] % 2 == 0) {
```

```
            pares++;
```

```
        } else {
```

```
            impares++;
```

```
        }
```

```
    }
```

```
    printf("quantidade de elementos pares: %d\n", pares);
```

```
    printf("quantidade de elementos ímpares: %d\n", impares);
```

```
    return 0;}
```

```
22         impares++;
23     }
24 }
25
26 printf("quantidade de elementos pares: %d\n", pares);
27 printf("quantidade de elementos ímpares: %d\n", impares);
28
29 return 0;
30 }
```

```
igite o numero de elementos do vetor: 5
digite os elementos do vetor:
```

```
2
4
6
7
8
```

```
quantidade de elementos pares: 4
quantidade de elementos ímpares: 1
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

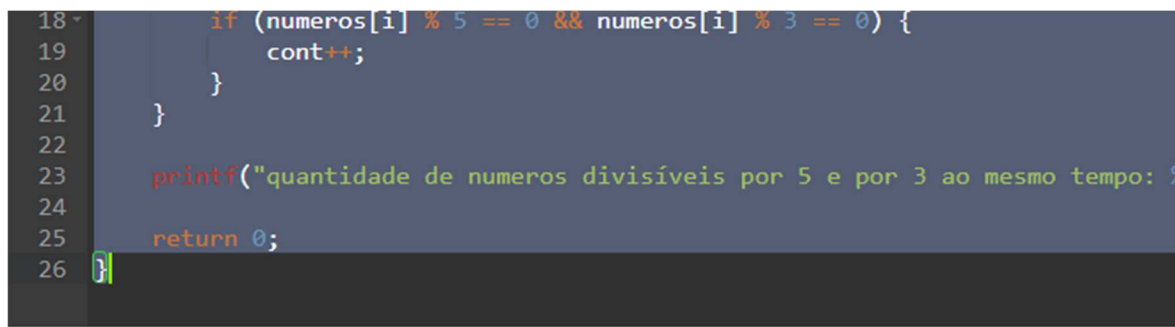
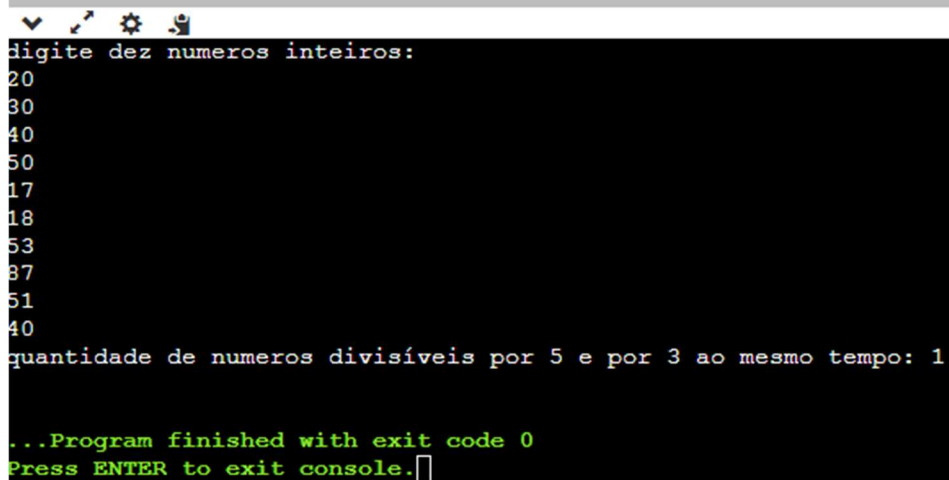
2). Desenvolver um algoritmo que leia dez números inteiro e verifique e imprima quantos são divisíveis por 5 e por 3 ao mesmo tempo.

```
#include <stdio.h>

#include <stdlib.h>

#include <stdio.h>

int main() {
    int numeros[10];
    int i, cont = 0;
    printf("digite dez numeros inteiros:\n");
    for (i = 0; i < 10; i++) {
        scanf("%d", &numeros[i]);
    }
    for (i = 0; i < 10; i++) {
        if (numeros[i] % 5 == 0 && numeros[i] % 3 == 0) {
            cont++;
        }
    }
    printf("quantidade de numeros divisíveis por 5 e por 3 ao mesmo tempo: %d\n", cont);
    return 0;}
```

A screenshot of a code editor with a dark background. The code is written in C and is color-coded. Line numbers 18 through 26 are visible on the left. The code includes headers, declares an array, reads input, and counts numbers divisible by both 5 and 3. The cursor is at the end of line 26.A screenshot of a terminal window with a black background. It shows the program's execution: a prompt to enter 10 integers, followed by the user entering 20, 30, 40, 50, 17, 18, 53, 37, 51, and 40. The program then outputs that 1 number is divisible by both 5 and 3. It ends with a message that the program finished with exit code 0 and a prompt to press ENTER to exit the console.

3). Fazer um programa que faz uma pesquisa com pessoas entre 18 e 80 anos. O programa deve solicitar a quantidade de pessoas a ser entrevistadas. Armazenar a idade dessas pessoas em um vetor e imprimir quantas pessoas de cada faixa etária foram entrevistadas de acordo com a tabela abaixo: ≥ 18 e < 35 \diamond jovem ≥ 35 e ≤ 65 \diamond adulto ≥ 65 e ≤ 80 \diamond idoso O programa deve imprimir o quantitativo de jovens, adultos e idosos. Desta forma essas variáveis que irão contar deverão ser inicializadas com zero

```
#include <stdio.h>

#include <stdlib.h>

#include <stdio.h>

int main() {

    int quantidade, i;

    int jovens = 0, adultos = 0, idosos = 0;

    int idade;

    printf("Digite a quantidade de pessoas a serem entrevistadas: ");

    scanf("%d", &quantidade);

    for (i = 0; i < quantidade; i++) {

        printf("Digite a idade da pessoa %d: ", i + 1);

        scanf("%d", &idade);

        if (idade >= 18 && idade < 35) {

            jovens++;

        } else if (idade >= 35 && idade < 65) {

            adultos++;

        } else if (idade >= 65 && idade <= 80) {

            idosos++;

        }

    }

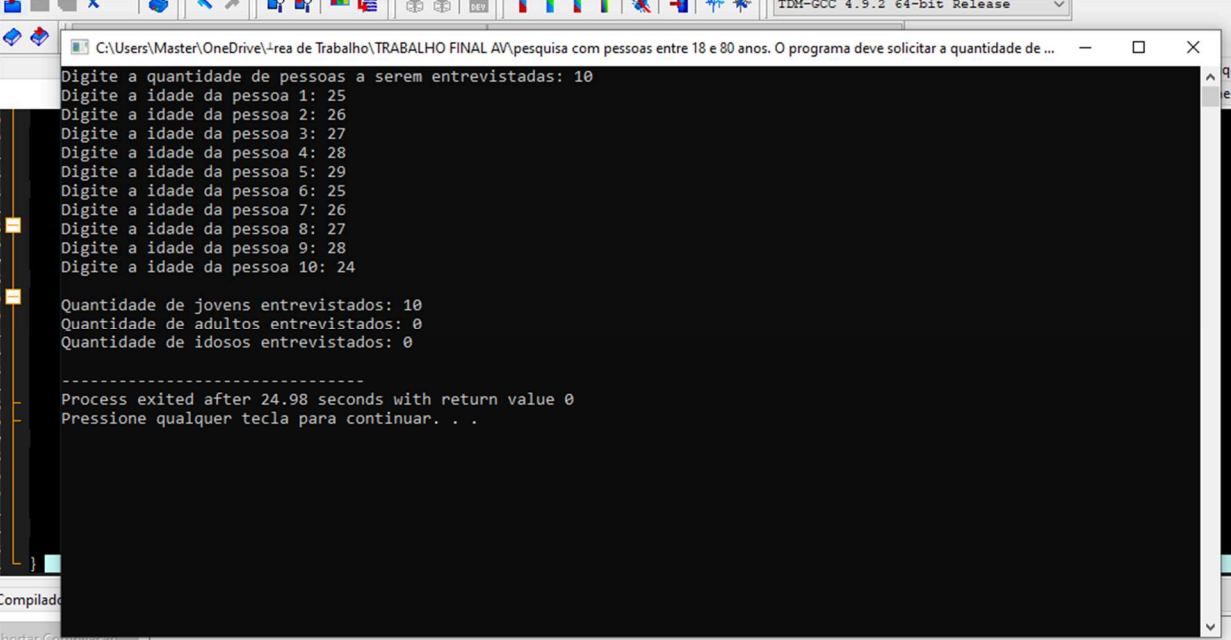
    printf("\nquantidade de jovens entrevistados: %d\n", jovens);

    printf("quantidade de adultos entrevistados: %d\n", adultos);

    printf("quantidade de idosos entrevistados: %d\n", idosos);

    return 0;

}
```



```
C:\Users\Master\OneDrive\rea de Trabalho\TRABALHO FINAL AV\pesquisa com pessoas entre 18 e 80 anos. O programa deve solicitar a quantidade de ...
Digite a quantidade de pessoas a serem entrevistadas: 10
Digite a idade da pessoa 1: 25
Digite a idade da pessoa 2: 26
Digite a idade da pessoa 3: 27
Digite a idade da pessoa 4: 28
Digite a idade da pessoa 5: 29
Digite a idade da pessoa 6: 25
Digite a idade da pessoa 7: 26
Digite a idade da pessoa 8: 27
Digite a idade da pessoa 9: 28
Digite a idade da pessoa 10: 24

Quantidade de jovens entrevistados: 10
Quantidade de adultos entrevistados: 0
Quantidade de idosos entrevistados: 0

-----
Process exited after 24.98 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

4) Faça um programa que leia 10 números inteiros, armazene-os em um vetor, solicite um valor de referência inteiro e: a) imprima os números do vetor que são maiores que o valor de referência b) retorne quantas vezes o valor de referência aparece no vetor

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

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

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

```
int main() {
```

```
    int numeros[10];
```

```
    int valor_referencia, i;
```

```
    int contador_maior = 0;
```

```
    int contador_referencia = 0;
```

```
    printf("Digite 10 números inteiros:\n");
```

```
    for (i = 0; i < 10; i++) {
```

```
        scanf("%d", &numeros[i]); }
```

```
    printf("Digite um valor de referencia inteiro: ");
```

```
    scanf("%d", &valor_referencia);
```

```
    printf("numeros do vetor maiores que o valor de referencia:\n");
```

```
    for (i = 0; i < 10; i++) {
```

```
        if (numeros[i] > valor_referencia) {
```

```
            printf("%d ", numeros[i]);
```

```
            contador_maior++;
```

```
        } if (numeros[i] == valor_referencia) {
```

```
            contador_referencia++;} }
```

```
    printf("\n");
```

```
    printf("O valor de referência aparece %d vezes no vetor.\n", contador_referencia);
```

```
    return 0;}
```

```
C:\Users\Master\OneDrive\Trabalho\TRABALHO FINAL AV\leia 10 n-meros inteiros, armazene-os em um vetor, solicite um valor.exe
Digite 10 n-meros inteiros:
1
2
3
4
5
6
7
7
7
7
Digite um valor de referencia inteiro: 7
N-meros do vetor maiores que o valor de referencia:
O valor de referencia aparece 3 vezes no vetor.
Process exited after 11.07 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Compilation results...

```
- Errors: 0
- Warnings: 0
```

5) fazer um programa que crie um Struct com a matricula, nome, nota da AV, simulado1 e simulado 2 de um aluno. O programa para cada aluno deve apresentar os dados lidos destes alunos na Struct. #include <stdio.h>

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

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

```
struct Aluno { int matricula;
```

```
    char nome[50];
```

```
    float nota_AV;
```

```
    float simulado1;
```

```
    float simulado2;};int main() {
```

```
    struct Aluno aluno;
```

```
    printf("Digite a matricula do aluno: ");
```

```
    scanf("%d", &aluno.matricula);
```

```
    printf("Digite o nome do aluno: ");
```

```
    scanf(" %[^\\n]s", aluno.nome);
```

```
    printf("Digite a nota da avaliacao: ");
```

```
    scanf("%f", &aluno.nota_AV);
```

```
    printf("Digite a nota do simulado 1: ");
```

```
    scanf("%f", &aluno.simulado1);
```

```
    printf("Digite a nota do simulado 2: ");
```

```
    scanf("%f", &aluno.simulado2);
```

```
    printf("\\nDados do aluno:\\n");
```

```
    printf("Matricula: %d\\n", aluno.matricula);
```

```
    printf("Nome: %s\\n", aluno.nome);
```

```
    printf("Nota da avaliacao: %.2f\\n", aluno.nota_AV);
```

```
    printf("Nota do simulado 1: %.2f\\n", aluno.simulado1);
```

```
    printf("Nota do simulado 2: %.2f\\n", aluno.simulado2); return 0;}
```

```
main.c
10 float simulado1;
11 float simulado2;
12 };
13
14 int main() {
15
16     struct Aluno aluno;
17
18     printf("Digite a matricula do aluno: ");
19     scanf("%d", &aluno.matricula);
20
21     printf("Digite o nome do aluno: ");
22     scanf(" %[^\\n]s", aluno.nome);
23
24     printf("Digite a nota da avaliacao: ");
25     scanf("%f", &aluno.nota);
26
27     printf("Digite a nota do simulado 1: ");
28     scanf("%f", &simulado1);
29
30     printf("Digite a nota do simulado 2: ");
31     scanf("%f", &simulado2);
32
33     printf("Dados do aluno:");
34     printf("Matricula: %d", aluno.matricula);
35     printf("Nome: %s", aluno.nome);
36     printf("Nota da avaliacao: %f", aluno.nota);
37     printf("Nota do simulado 1: %f", simulado1);
38     printf("Nota do simulado 2: %f", simulado2);
39
40     return 0;
41 }
```

input

```
Digite a matricula do aluno: 202402370014
Digite o nome do aluno: Renato de souza de freitas
Digite a nota da avaliacao: 9.5
Digite a nota do simulado 1: 5.6
Digite a nota do simulado 2: 9.6

Dados do aluno:
Matricula: 538907102
Nome: Renato de souza de freitas
Nota da avaliacao: 9.50
Nota do simulado 1: 5.60
Nota do simulado 2: 9.60

...Program finished with exit code 0
Press ENTER to exit console.
```

6) fazer um programa que solicite ao usuário na mais um número para calcular a tabuada. Crie uma função que faz o cálculo da tabuada. (Fonte de estudo slide 56). OBS: a única diferença é que a geração da tabuada tem que ser em função.

7) Criar um programa em linguagem C que calcule o Índice de Massa Corporal (IMC) de um indivíduo e classifique sua situação de acordo com os seguintes parâmetros: • Abaixo do peso: IMC menor que 18,5 • Peso normal: IMC entre 18,5 e 24,9 • Sobrepeso: IMC entre 25 e 29,9 • Obesidade grau I: IMC entre 30 e 34,9 • Obesidade grau II: IMC entre 35 e 39,9 • Obesidade grau III: IMC maior ou igual a 40 O programa deve solicitar ao usuário que digite seu peso em quilogramas e sua altura em metros. Em seguida, o programa deve calcular o IMC utilizando a seguinte fórmula: $IMC = \text{peso} / (\text{altura} * \text{altura})$

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

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

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

```
float calcularIMC(float peso, float altura) {
```

```
    return peso / (altura * altura);}
```

```
void classificarIMC(float imc) {
```

```
    printf("Seu IMC é %.2f. ", imc);
```

```
    if (imc < 18.5) {
```

```
        printf("voce estq abaixo do peso.\n");
```

```
    } else if (imc >= 18.5 && imc < 25) {
```

```
        printf("voce esta com peso normal.\n");
```

```
    } else if (imc >= 25 && imc < 30) {
```

```
        printf("voce esta com sobrepeso.\n");
```

```
    } else if (imc >= 30 && imc < 35) {
```

```
        printf("voce esta com obesidade grau 1.\n");
```

```
    } else if (imc >= 35 && imc < 40) {
```

```
        printf("voce esta com obesidade grau 2.\n");
```

```
    } else {
```

```
        printf("voce esta com obesidade grau 3.\n");
```

```
    }
```

```
}
```

```
int main() {
```

```
    float peso, altura;
```

```
    printf("digite seu peso (em kg): ");
```

```
    scanf("%f", &peso);
```

```
    printf("digite sua altura (em metros): ");
```

```
    scanf("%f", &altura);
```

```
    float imc = calcularIMC(peso, altura);
```

```
    classificarIMC(imc);
```

```
    return 0;
```



```
29     printf("digite seu peso (em kg): ");
30     scanf("%f", &peso);
31
32     printf("digite sua altura (em metros): ");
33     scanf("%f", &altura);
34
35     float imc = calcularIMC(peso, altura);
36     classificarIMC(imc);
37     return 0;
38 }
39
```

digite seu peso (em kg): 138
digite sua altura (em metros): 1.84
Seu IMC é 40.76. voce esta com obesidade grau 3.

...Program finished with exit code 0
Press ENTER to exit console.