**2º LISTA DE EXERCÍCIOS - INTRODUÇÃO À PROGRAMAÇÃO**

**Nome: Vinicius de Souza Paixão**

**1)**

#include <stdio.h>

#include <stdlib.h>

int main (){

int i;

int vet[15];

printf("Informe 15 numeros inteiros: \n");

for(i=0;i<15;i++){

printf("Informe um numero: ");

scanf("%d", &vet[i]);

}

printf("Os numeros informados foram: \n");

for(i=0;i<15;i++){

printf("%i", vet[i]);

printf("\n");

}

return(0);

}

**2)**

#include <stdio.h>

#include <stdlib.h>

int main (){

int i;

int vet[15];

printf("Informe 15 numeros inteiros: \n");

for(i=0;i<15;i++){

printf("Informe um numero: ");

scanf("%d", &vet[i]);

}

printf("Os numeros informados foram: \n");

for(i=0;i<15;i++){

printf("%i", vet[i]);

printf("\n");

}

return(0);

**3)**

#include <stdio.h>

#include <stdlib.h>

int main (){

int vet[12] = {2,3,6,9,12,15,18,21,24,27,30,33};

int i;

int dvet[12];

int vetres[12];

int verificapar;

for(i=0;i<12;i++) {

verificapar = vet[i] % 2;

if (verificapar == 0){

vetres[i] = vet[i] / 2;

}

else {

vetres[i] = vet[i] \* 3;

}

}

printf("Valores pares: \n");

for(i=0;i<12;i++){

printf("%d", vetres[i]);

printf("\n");

}

return(0);

}

**4)**

#include <stdio.h>

#include <stdlib.h>

int main (){

char nomes[10][20] = {"lorena", "matheus", "viviane", "milena", "marjory", "vinicius", "joao", "hanna", "liam", "athena"};

char nomeb[10];

int i, busca;

printf("Procure um nome: \n");

scanf("%s", nomeb);

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

if(strcmp(nomeb, nomes[i]) == 0){

busca = 1;

}

}

if(busca == 1){

printf("ACHEI");

}

else {

printf("NAO ACHEI");

}

return(0);

}

**5)**

#include <stdio.h>

#include <stdlib.h>

int main() {

int vet1[20] = {2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40}, vet2[20] = {3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,60};

int vetdif[20], vetsoma[20], vetmult[20];

int i;

for(i=0;i<20;i++){

vetdif[i] = vet2[i] - vet1[i];

vetsoma[i] = vet2[i] + vet1[i];

vetmult [i] = vet2[i] \* vet1[i];

}

printf("Vetor formado pela difereça dos dois vetores lidos: ");

for(i=0;i<20;i++){

printf("%d", vetdif[i]);

printf("\n");

}

printf("Vetor formado pela soma dos dois vetores lidos: ");

for(i=0;i<20;i++){

printf("%d", vetsoma[i]);

printf("\n");

}

printf("Vetor formado pela multiplicação dos dois vetores lidos: ");

for(i=0;i<20;i++){

printf("%d", vetmult[i]);

printf("\n");

}

**6)**

#include <stdio.h>

#include<stdlib.h>

int main (){

int vetor[], int tamanho)

int i, j, temp;

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

for (j = i + 1; j < tamanho; j++) {

if (vetor[i] % 2 == 0 && vetor[j] % 2 == 0 && vetor[i] > vetor[j]) {

temp = vetor[i];

vetor[i] = vetor[j];

vetor[j] = temp;

}

}

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

for (j = i + 1; j < tamanho; j++) {

if (vetor[i] % 2 != 0 && vetor[j] % 2 != 0 && vetor[i] < vetor[j]) {

temp = vetor[i];

vetor[i] = vetor[j];

vetor[j] = temp;

}

}

}

int main() {

int tamanho;

printf("Digite o tamanho do vetor: ");

scanf("%d", &tamanho);

int vetor[tamanho];

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

for (int i = 0; i < tamanho; i++) {

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

}

organizarNumeros(vetor, tamanho);

printf("Vetor organizado:\n");

for (int i = 0; i < tamanho; i++) {

printf("%d ", vetor[i]);

}

return 0;

}

return 0;

}

**7)**

#include <stdio.h>

#include <stdlib.h>

int vetores\_sao\_iguais(int \*vet1, int \*vet2, int tam) {

for (int i = 0; i < tam; i++) {

if (vet1[i] != vet2[i]) {

return 0;

}

}

return 1;

}

int main() {

int tam1, tam2;

printf("Digite o tamanho do primeiro vetor: \n");

scanf("%d", &tam1);

printf("Digite o tamanho do segundo vetor: \n");

scanf("%d", &tam2);

if (tam1 != tam2) {

printf("Os vetores sao diferentes.\n");

return 0;

}

int vet1[tam1], vet2[tam2];

printf("Digite o conteudo do primeiro vetor: \n");

for (int i = 0; i < tam1; i++) {

scanf("%d", &vet1[i]);

}

printf("Digite o conteudo do segundo vetor: \n");

for (int i = 0; i < tam2; i++) {

scanf("%d", &vet2[i]);

}

if (vetores\_sao\_iguais(vet1, vet2, tam1)) {

printf("Os vetores sao iguais.\n");

} else {

printf("Os vetores sao diferentes.\n");

}

return 0;

}

**MATRIZES**

**1)** #include <stdio.h>

#include<stdlib.h>

int main() {

int MAT[10][15];

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

for (int j = 0; j < 15; j++) {

MAT[i][j] = i + j;

}

}

printf("Soma e paridade das linhas:\n");

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

int somaLinha = 0;

for (int j = 0; j < 15; j++) {

somaLinha += MAT[i][j];

}

printf("Linha %d: Soma = %d, par ou impar = %s\n", i + 1, somaLinha, (somaLinha % 2 == 0) ? "Par" : "impar");

}

printf("\nSoma e paridade das colunas:\n");

for (int j = 0; j < 15; j++) {

int somaColuna = 0;

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

somaColuna += MAT[i][j];

}

printf("Coluna %d: Soma = %d, par ou impar = %s\n", j + 1, somaColuna, (somaColuna % 2 == 0) ? "Par" : "impar");

}

return 0;

}

**2)**

#include <stdio.h>

#include <stdlib.h>

#define TAMANHO\_MATRIZ 50

int main() {

float matriz[TAMANHO\_MATRIZ][TAMANHO\_MATRIZ];

)

for (int i = 0; i < TAMANHO\_MATRIZ; i++) {

for (int j = 0; j < TAMANHO\_MATRIZ; j++) {

matriz[i][j] = i \* TAMANHO\_MATRIZ + j + 1;

}

}

float diagonalPrincipal[TAMANHO\_MATRIZ];

for (int i = 0; i < TAMANHO\_MATRIZ; i++) {

diagonalPrincipal[i] = matriz[i][i];

}

printf("Vetor da diagonal principal:\n");

for (int i = 0; i < TAMANHO\_MATRIZ; i++) {

printf("%.2f ", diagonalPrincipal[i]);

}

return 0;

}

**3)**

#include <stdio.h>

#include<stdlib.h>

void gerarTransposta(int matriz[][100], int transposta[][100], int linhas, int colunas) {

for (int i = 0; i < linhas; i++) {

for (int j = 0; j < colunas; j++) {

transposta[j][i] = matriz[i][j];

}

}

}

void imprimirMatriz(int matriz[][100], int linhas, int colunas) {

for (int i = 0; i < linhas; i++) {

for (int j = 0; j < colunas; j++) {

printf("%d\t", matriz[i][j]);

}

printf("\n");

}

}

int main() {

int linhas, colunas;

printf("Digite o número de linhas da matriz: ");

scanf("%d", &linhas);

printf("Digite o número de colunas da matriz: ");

scanf("%d", &colunas);

int matriz[linhas][100];

int transposta[100][linhas];

printf("Digite os elementos da matriz:\n");

for (int i = 0; i < linhas; i++) {

for (int j = 0; j < colunas; j++) {

scanf("%d", &matriz[i][j]);

}

}

printf("\nMatriz original:\n");

imprimirMatriz(matriz, linhas, colunas);

gerarTransposta(matriz, transposta, linhas, colunas);

printf("\nMatriz transposta:\n");

imprimirMatriz(transposta, colunas, linhas);

return 0;

}

**3)**

#include <stdio.h>

void printMatrix(int mat[][100], int lin, int col) {

for (int i = 0; i < lin; i++) {

for (int j = 0; j < col; j++) {

printf("%d ", mat[i][j]);

}

printf("\n");

}

}

void transposeMatrix(int mat[][100], int lin, int col, int resul[][100]) {

for (int i = 0; i < lin; i++) {

for (int j = 0; j < col; j++) {

resul[j][i] = mat[i][j];

}

}

}

int main() {

int linhas, colunas;

printf("Digite o número de linhas da matriz: ");

scanf("%d", &linhas);

printf("Digite o número de colunas da matriz: ");

scanf("%d", &colunas);

int matriz[linhas][colunas];

int matrizTransposta[colunas][linhas];

printf("Digite os elementos da matriz:\n");

for (int i = 0; i < linhas; i++) {

for (int j = 0; j < colunas; j++) {

printf("Elemento [%d][%d]: ", i+1, j+1);

scanf("%d", &matriz[i][j]);

}

}

printf("\nMatriz original:\n");

printMatrix(matriz, linhas, colunas);

transposeMatrix(matriz, linhas, colunas, matrizTransposta);

printf("\nMatriz transposta:\n");

printMatrix(matrizTransposta, colunas, linhas);

return 0;

}

**4)**

#include <stdio.h>

#include <conio.h>

main() {

int i, j, linhaA, colunaA, linhaB, colunaB, x;

printf("Informe a quntidade de linhas da matriz A : ");

scanf("%d",&linhaA);

printf("Informe a quantidade de colunas da matriz A : ");

scanf("%d",&colunaA);

printf("Informe a quntidade de linhas da matriz B : ");

scanf("%d",&linhaB);

printf("Informe a quantidade de colunas da matriz B : ");

scanf("%d",&colunaB);

float matrizA[linhaA][colunaA], matrizB[linhaB][colunaB], matrizC[linhaA][colunaB], aux = 0;

if(colunaA == linhaB) {

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

for(j = 0; j < colunaA; j++) {

printf("\n\n Informe o valor da %d%c Linha e da %d%c Coluna da Matriz A: ", i+1, 167, j+1, 167);

scanf("%f", &matrizA[i][j]);

}

}

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

for(j = 0; j < colunaB; j++) {

printf("\n\n Informe o valor da %d%c Linha e da %d%c Coluna da 2%c Matriz B: ", i+1, 167, j+1, 167, 167);

scanf("%f", &matrizB[i][j]);

}

}

printf("Matriz Gerada");

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

for(j = 0; j < colunaA; j++) {

printf("%6.f", matrizA[i][j]);

}

}

printf("Matriz Gerada ");

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

for(j = 0; j < colunaB; j++) {

printf("%6.f", matrizB[i][j]);

}

}

printf(" Matriz Gerada");

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

for(j = 0; j < colunaB; j++) {

matrizC[i][j] = 0;

for(x = 0; x < linhaB; x++) {

aux += matrizA[i][x] \* matrizB[x][j];

}

matrizC[i][j] = aux;

aux = 0;

}

}

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

for(j = 0; j < colunaB; j++) {

printf("%6.f", matrizC[i][j]);

}

printf("\n");

}

printf("\n");

} else {

printf("\n Nao ha com multiplicar as matrizes dadas ");

}

return 0;

}

**5)**

#include <stdio.h>

int main(){

int m, n;

printf("Quantas linhas vai ter cada matriz? ");

scanf("%d", &m);

printf("Quantas colunas vai ter cada matriz? ");

scanf("%d", &n);

int a[m][n], b[m][n], c[m][n];

printf("Digite os valores da matriz A:\n");

for (int i=0; i<m; i++) {

for (int j=0; j<n; j++) {

printf("Elemento [%d,%d]: ", i, j);

scanf("%d", &a[i][j]);

}

}

printf("Digite os valores da matriz B:\n");

for (int i=0; i<m; i++) {

for (int j=0; j<n; j++) {

printf("Elemento [%d,%d]: ", i, j);

scanf("%d", &b[i][j]);

}

}

for (int i=0; i<m; i++) {

for (int j=0; j<n; j++) {

c[i][j] = a[i][j] + b[i][j];

}

}

printf("MATRIZ SOMA:\n");

for (int i=0; i<m; i++) {

for (int j=0; j<n; j++) {

printf("%d ", c[i][j]);

}

printf("\n");

}

return 0;

}

**6)**

#include <stdio.h>

#define ORDEM 4

void imprimirMatriz(int mat[][ORDEM]) {

printf("Matriz:\n");

for (int i = 0; i < ORDEM; i++) {

for (int j = 0; j < ORDEM; j++) {

printf("%d ", mat[i][j]);

}

printf("\n");

}

}

int somarQuadradosPrimeiraColuna(int mat[][ORDEM]) {

int soma = 0;

for (int i = 0; i < ORDEM; i++) {

soma += mat[i][0] \* mat[i][0];

}

return soma;

}

int somarTerceiraLinha(int mat[][ORDEM]) {

int soma = 0;

for (int j = 0; j < ORDEM; j++) {

soma += mat[2][j];

}

return soma;

}

int somarDiagonalPrincipal(int mat[][ORDEM]) {

int soma = 0;

for (int i = 0; i < ORDEM; i++) {

soma += mat[i][i];

}

return soma;

}

int somarElementosIndiceParSegundaLinha(int mat[][ORDEM]) {

int soma = 0;

for (int j = 0; j < ORDEM; j += 2) {

soma += mat[1][j];

}

return soma;

}

int main() {

int matriz[ORDEM][ORDEM];

printf("Digite os elementos da matriz de ordem %d:\n", ORDEM);

for (int i = 0; i < ORDEM; i++) {

for (int j = 0; j < ORDEM; j++) {

printf("Elemento [%d][%d]: ", i + 1, j + 1);

scanf("%d", &matriz[i][j]);

}

}

imprimirMatriz(matriz);

int somaQuadradosPrimeiraColuna = somarQuadradosPrimeiraColuna(matriz);

printf("\nb) Soma dos quadrados da primeira coluna: %d\n", somaQuadradosPrimeiraColuna);

int somaTerceiraLinha = somarTerceiraLinha(matriz);

printf("c) Soma da terceira linha: %d\n", somaTerceiraLinha);

int somaDiagonalPrincipal = somarDiagonalPrincipal(matriz);

printf("d) Soma da diagonal principal: %d\n", somaDiagonalPrincipal);

int somaElementosIndiceParSegundaLinha = somarElementosIndiceParSegundaLinha(matriz);

printf("e) Soma dos elementos de índice par da segunda linha: %d\n", somaElementosIndiceParSegundaLinha);

return 0;

}

**STRUCT  
1)**

#include <stdio.h>

#include <string.h>

struct Pessoa {

char nome[50];

char endereco[100];

char telefone[15];

};

int compararPessoas(const void \*a, const void \*b) {

return strcmp(((struct Pessoa\*)a)->nome, ((struct Pessoa\*)b)->nome);

}

int main() {

Pessoa para armazenar as informações

struct Pessoa pessoas[10];

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

printf("Digite o nome da pessoa %d: ", i + 1);

fgets(pessoas[i].nome, sizeof(pessoas[i].nome), stdin);

pessoas[i].nome[strcspn(pessoas[i].nome, "\n")] = '\0'

printf("Digite o endereço da pessoa %d: ", i + 1);

fgets(pessoas[i].endereco, sizeof(pessoas[i].endereco), stdin);

pessoas[i].endereco[strcspn(pessoas[i].endereco, "\n")] = '\0';

printf("Digite o telefone da pessoa %d: ", i + 1);

fgets(pessoas[i].telefone, sizeof(pessoas[i].telefone), stdin);

pessoas[i].telefone[strcspn(pessoas[i].telefone, "\n")] = '\0';

printf("\n");

}

qsort(pessoas, 10, sizeof(struct Pessoa), compararPessoas);

printf("Pessoas em ordem alfabética:\n");

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

printf("Nome: %s\n", pessoas[i].nome);

printf("Endereço: %s\n", pessoas[i].endereco);

printf("Telefone: %s\n", pessoas[i].telefone);

printf("\n");

}

return 0;

}

**2)**

#include <stdio.h>

struct Aluno {

int matricula;

char nome[50];

int codigoDisciplina;

float nota1;

float nota2;

};

int main() {

struct Aluno alunos[10];

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

printf("Digite a matrícula do aluno %d: ", i + 1);

scanf("%d", &alunos[i].matricula);

printf("Digite o nome do aluno %d: ", i + 1);

scanf("%s", alunos[i].nome);

printf("Digite o código da disciplina do aluno %d: ", i + 1);

scanf("%d", &alunos[i].codigoDisciplina);

printf("Digite a nota 1 do aluno %d: ", i + 1);

scanf("%f", &alunos[i].nota1);

printf("Digite a nota 2 do aluno %d: ", i + 1);

scanf("%f", &alunos[i].nota2);

printf("\n");

}

printf("Listagem final dos alunos com médias finais:\n");

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

float mediaFinal = (alunos[i].nota1 + 2 \* alunos[i].nota2) / 3.0;

printf("Matrícula: %d\n", alunos[i].matricula);

printf("Nome: %s\n", alunos[i].nome);

printf("Código da Disciplina: %d\n", alunos[i].codigoDisciplina);

printf("Nota 1: %.2f\n", alunos[i].nota1);

printf("Nota 2: %.2f\n", alunos[i].nota2);

printf("Média Final: %.2f\n", mediaFinal);

printf("\n");

}

return 0;

}

**3)**

#include <stdio.h>

struct Horario {

int hora;

int minutos;

int segundos;

};

struct Data {

int dia;

int mes;

int ano;

};

struct Compromisso {

struct Data dataCompromisso;

struct Horario horarioCompromisso;

char texto[100];

};

int main() {

struct Horario horarioExemplo = {12, 30, 45};

struct Data dataExemplo = {24, 11, 2023};

struct Compromisso compromissoExemplo = {{24, 11, 2023}, {12, 30, 45}, "Reunião de equipe"};

printf("Exemplo de Horario: %02d:%02d:%02d\n", horarioExemplo.hora, horarioExemplo.minutos, horarioExemplo.segundos);

printf("Exemplo de Data: %02d/%02d/%04d\n", dataExemplo.dia, dataExemplo.mes, dataExemplo.ano);

printf("Exemplo de Compromisso:\n");

printf("Data: %02d/%02d/%04d\n", compromissoExemplo.dataCompromisso.dia, compromissoExemplo.dataCompromisso.mes, compromissoExemplo.dataCompromisso.ano);

printf("Horario: %02d:%02d:%02d\n", compromissoExemplo.horarioCompromisso.hora, compromissoExemplo.horarioCompromisso.minutos, compromissoExemplo.horarioCompromisso.segundos);

printf("Texto: %s\n", compromissoExemplo.texto);

return 0;

}

**4)**

#include <stdio.h>

struct Pessoa {

char nome[50];

int idade;

char endereco[100];

};

int main() {

struct Pessoa pessoa;

printf("Digite o nome da pessoa: ");

scanf("%s", pessoa.nome);

printf("Digite a idade da pessoa: ");

scanf("%d", &pessoa.idade);

printf("Digite o endereço da pessoa: ");

scanf("%s", pessoa.endereco);

printf("\nDados armazenados na estrutura:\n");

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

printf("Idade: %d\n", pessoa.idade);

printf("Endereço: %s\n", pessoa.endereco);

return 0;

}

**5)**

#include <stdio.h>

struct Aluno {

char nome[50];

int matricula;

char curso[50];

};

int main() {

struct Aluno alunos[5];

for (int i = 0; i < 5; i++) {

printf("Digite o nome do aluno %d: ", i + 1);

scanf("%s", alunos[i].nome);

printf("Digite o número de matrícula do aluno %d: ", i + 1);

scanf("%d", &alunos[i].matricula);

printf("Digite o curso do aluno %d: ", i + 1);

scanf("%s", alunos[i].curso);

printf("\n");

}

printf("Dados dos alunos:\n");

for (int i = 0; i < 5; i++) {

printf("Aluno %d:\n", i + 1);

printf("Nome: %s\n", alunos[i].nome);

printf("Número de Matrícula: %d\n", alunos[i].matricula);

printf("Curso: %s\n", alunos[i].curso);

printf("\n");

}

return 0;

}

**6)**

#include <stdio.h>

struct Aluno {

int matricula;

char nome[50];

float nota1;

float nota2;

float nota3;

};

int main() {

struct Aluno alunos[5];

for (int i = 0; i < 5; i++) {

printf("Digite a matrícula do aluno %d: ", i + 1);

scanf("%d", &alunos[i].matricula);

printf("Digite o nome do aluno %d: ", i + 1);

scanf("%s", alunos[i].nome);

printf("Digite a nota da primeira prova do aluno %d: ", i + 1);

scanf("%f", &alunos[i].nota1);

printf("Digite a nota da segunda prova do aluno %d: ", i + 1);

scanf("%f", &alunos[i].nota2);

printf("Digite a nota da terceira prova do aluno %d: ", i + 1);

scanf("%f", &alunos[i].nota3);

printf("\n");

}

float maiorNota1 = alunos[0].nota1;

int indiceMaiorNota1 = 0;

for (int i = 1; i < 5; i++) {

if (alunos[i].nota1 > maiorNota1) {

maiorNota1 = alunos[i].nota1;

indiceMaiorNota1 = i;

}

}

float maiorMedia = (alunos[0].nota1 + alunos[0].nota2 + alunos[0].nota3) / 3.0;

float menorMedia = maiorMedia;

int indiceMaiorMedia = 0;

int indiceMenorMedia = 0;

for (int i = 1; i < 5; i++) {

float media = (alunos[i].nota1 + alunos[i].nota2 + alunos[i].nota3) / 3.0;

if (media > maiorMedia) {

maiorMedia = media;

indiceMaiorMedia = i;

}

if (media < menorMedia) {

menorMedia = media;

indiceMenorMedia = i;

}

}

printf("Aluno com maior nota na primeira prova:\n");

printf("Nome: %s\n", alunos[indiceMaiorNota1].nome);

printf("Nota: %.2f\n\n", alunos[indiceMaiorNota1].nota1);

printf("Aluno com maior média geral:\n");

printf("Nome: %s\n", alunos[indiceMaiorMedia].nome);

printf("Média: %.2f\n\n", maiorMedia);

printf("Aluno com menor média geral:\n");

printf("Nome: %s\n", alunos[indiceMenorMedia].nome);

printf("Média: %.2f\n\n", menorMedia);

printf("Situação de aprovação dos alunos:\n");

for (int i = 0; i < 5; i++) {

float media = (alunos[i].nota1 + alunos[i].nota2 + alunos[i].nota3) / 3.0;

printf("Aluno %d (%s): ", i + 1, alunos[i].nome);

if (media >= 6.0) {

printf("Aprovado\n");

} else {

printf("Reprovado\n");

}

}

return 0;

}

**RECURSIVIDADE**

**1)**

#include <stdio.h>

int calcularFatorial(int n);

int main() {

int numero;

printf("Digite um número para calcular o fatorial: ");

scanf("%d", &numero);

int resultado = calcularFatorial(numero);

printf("O fatorial de %d é: %d\n", numero, resultado);

return 0;

}

int calcularFatorial(int n) {

if (n == 0) {

return 1;

} else {

return n \* calcularFatorial(n - 1);

}

}

**2)**

#include <stdio.h>

float calcularSoma(float num1, float num2);

float calcularMultiplicacao(float num1, float num2);

float calcularSubtracao(float num1, float num2);

float calcularDivisao(float num1, float num2);

int main() {

float numero1, numero2;

printf("Digite o primeiro número: ");

scanf("%f", &numero1);

printf("Digite o segundo número: ");

scanf("%f", &numero2);

printf("Soma: %.2f\n", calcularSoma(numero1, numero2));

printf("Multiplicação: %.2f\n", calcularMultiplicacao(numero1, numero2));

printf("Subtração: %.2f\n", calcularSubtracao(numero1, numero2));

printf("Divisão: %.2f\n", calcularDivisao(numero1, numero2));

return 0;

}

float calcularSoma(float num1, float num2) {

return num1 + num2;

}

float calcularMultiplicacao(float num1, float num2) {

if (num2 == 0) {

return 0;

} else {

return num1 + calcularMultiplicacao(num1, num2 - 1);

}

}

float calcularSubtracao(float num1, float num2) {

return num1 - num2;

}

float calcularDivisao(float num1, float num2) {

if (num2 != 0) {

return num1 / num2;

} else {

printf("Erro: Divisão por zero não é permitida.\n");

return 0;

}

}

**3)**

#include <stdio.h>

int contarDigitos(int numero);

int main() {

int numero;

printf("Digite um número: ");

scanf("%d", &numero);

int numeroDeDigitos = contarDigitos(numero);

printf("O número digitado possui %d dígitos.\n", numeroDeDigitos);

return 0;

}

int contarDigitos(int numero) {

if (numero == 0) {

return 1;

} else {

return 1 + contarDigitos(numero / 10);

}

}

**4)**

#include <stdio.h>

#include <string.h>

void inverterString(char str[], int inicio, int fim);

int main() {

char string[100];

printf("Digite uma string: ");

fgets(string, sizeof(string), stdin);

string[strcspn(string, "\n")] = '\0';

inverterString(string, 0, strlen(string) - 1);

printf("String Invertida: %s\n", string);

return 0;

}

void inverterString(char str[], int inicio, int fim) {

if (inicio < fim) {

char temp = str[inicio];

str[inicio] = str[fim];

str[fim] = temp;

inverterString(str, inicio + 1, fim - 1);

}

}

**5)**

#include <stdio.h>

int calcularPotencia(int a, int b);

int main() {

int base, expoente;

printf("Digite a base: ");

scanf("%d", &base);

printf("Digite o expoente: ");

scanf("%d", &expoente);

int resultado = calcularPotencia(base, expoente);

printf("%d elevado a %d é: %d\n", base, expoente, resultado);

return 0;

}

int calcularPotencia(int a, int b) {

if (b == 0) {

return 1;

} else {

return a \* calcularPotencia(a, b - 1);

}

}

**CONDICIONAIS LÓGICOS**

**1)**

#include <stdio.h>

void fazerCheckIn() {

char nome[50], cpf[15], telefone[15], cidade[50], estado[3];

printf("Informe seu nome: ");

scanf("%s", nome);

printf("Informe seu CPF: ");

scanf("%s", cpf);

printf("Informe seu telefone: ");

scanf("%s", telefone);

printf("Informe sua cidade: ");

scanf("%s", cidade);

printf("Informe seu estado: ");

scanf("%s", estado);

printf("\nCheck-in realizado com sucesso!\n");

}

void chamarServicoQuarto() {

char nome[50];

int numeroQuarto;

printf("Informe seu nome: ");

scanf("%s", nome);

printf("Informe o número do quarto: ");

scanf("%d", &numeroQuarto);

printf("\nServiço de quarto chamado para o quarto %d.\n", numeroQuarto);

}

void fazerPedido() {

int opcao;

printf("\nMenu de Opções:\n");

printf("1. Comida\n");

printf("2. Bebida\n");

printf("3. Finalizar Pedido\n");

printf("Escolha uma opção: ");

scanf("%d", &opcao);

switch (opcao) {

case 1:

printf("Pedido de comida realizado!\n");

break;

case 2:

printf("Pedido de bebida realizado!\n");

break;

case 3:

printf("Pedido finalizado. Obrigado!\n");

break;

default:

printf("Opção inválida.\n");

}

}

int main() {

int escolha;

printf("Menu de Opções:\n");

printf("1. Fazer Check-in\n");

printf("2. Chamar Serviço de Quarto\n");

printf("3. Fazer Pedido\n");

printf("Escolha uma opção: ");

scanf("%d", &escolha);

switch (escolha) {

case 1:

fazerCheckIn();

break;

case 2:

chamarServicoQuarto();

break;

case 3:

fazerPedido();

break;

default:

printf("Opção inválida.\n");

}

return 0;

}

**2)**

#include <stdio.h>

#include <math.h>

int main() {

int numeroLados;

float medidaLado, area;

printf("Digite o número de lados do polígono: ");

scanf("%d", &numeroLados);

printf("Digite a medida do lado (em cm): ");

scanf("%f", &medidaLado);

if (numeroLados == 3) {

float s = 3 \* medidaLado / 2; /

area = sqrt(s \* (s - medidaLado) \* (s - medidaLado) \* (s - medidaLado));

printf("TRIÂNGULO\n");

printf("Área: %.2f cm²\n", area);

} else if (numeroLados == 4) {

area = medidaLado \* medidaLado;

printf("QUADRADO\n");

printf("Área: %.2f cm²\n", area);

} else if (numeroLados == 5) {

printf("PENTÁGONO\n");

} else {

printf("Número de lados inválido. Este programa trata apenas triângulos, quadrados e pentágonos.\n");

}

return 0;

}

**LAÇOS DE REPETIÇÃO1)**

#include <stdio.h>

int main() {

int numero, soma = 0, quantidade = 0;

do {

printf("Digite um número (ou 0 para sair): ");

scanf("%d", &numero);

if (numero % 2 == 0 && numero != 0) {

soma += numero;

quantidade++;

}

} while (numero != 0);

if (quantidade > 0) {

float media = (float)soma / quantidade;

printf("A média dos números pares é: %.2f\n", media);

} else {

printf("Nenhum número par foi digitado.\n");

}

return 0;

}

**2)**

#include <stdio.h>

int main() {

int numero, soma = 0, quantidade = 0;

do {

printf("Digite um número positivo (ou negativo para sair): ");

scanf("%d", &numero);

if (numero >= 0) {

soma += numero;

quantidade++;

}

} while (numero >= 0);

if (quantidade > 0) {

float media = (float)soma / quantidade;

printf("A média dos números positivos é: %.2f\n", media);

} else {

printf("Nenhum número positivo foi digitado.\n");

}

return 0;

}

**3)**

#include <stdio.h>

int main() {

int codigoVoto, votosCandidatos[4] = {0}, votosNulos = 0, votosBrancos = 0;

printf("Digite o código do candidato (ou 0 para encerrar): ");

while (1) {

scanf("%d", &codigoVoto);

if (codigoVoto == 0) {

break; // Encerra o loop quando o código do voto é 0

}

switch (codigoVoto) {

case 1:

case 2:

case 3:

case 4:

votosCandidatos[codigoVoto - 1]++;

break;

case 5:

votosNulos++;

break;

case 6:

votosBrancos++;

break;

default:

printf("Código de voto inválido.\n");

}

}

printf("\nTotal de votos para cada candidato:\n");

for (int i = 0; i < 4; i++) {

printf("Candidato %d: %d votos\n", i + 1, votosCandidatos[i]);

}

printf("Total de votos nulos: %d\n", votosNulos);

printf("Total de votos em branco: %d\n", votosBrancos);

return 0;

}

**4)**

#include <stdio.h>

int main() {

int valor, contadorNegativos = 0;

for (int i = 0; i < 5; i++) {

printf("Digite o valor %d: ", i + 1);

scanf("%d", &valor);

if (valor < 0) {

contadorNegativos++;

}

}

printf("Quantidade de valores negativos: %d\n", contadorNegativos);

return 0;