**LISTA 3 - ATIVIDADE PRÁTICA III – VALOR 20 PONTOS**

Nome: Rafael Akashi França- Ciência da Computação - 2 BN

**EXERCICIO 1**

#include <stdio.h>

//rafael akashi

typedef struct {

double peso;

int idade;

double altura;

} pessoa; // Definimos uma struct chamada pessoa que contém as variáveis peso, idade e altura.

void coletarDados(pessoa pessoas[], int numPessoas) {

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

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

scanf("%d", &pessoas[i].idade);

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

scanf("%lf", &pessoas[i].altura);

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

scanf("%lf", &pessoas[i].peso);

} // Função que solicita e coleta os dados de cada pessoa no array.

}

int main() {

pessoa pessoas[5], p1; // Declara um array de structs 'pessoas' e uma variável struct 'p1'.

pessoas[0].peso = 80.6; // Atribui um valor ao campo 'peso' da primeira posição do array.

pessoas[0].idade = 40; // Atribui um valor ao campo 'idade' da primeira posição do array.

pessoas[0].altura = 1.70; // Atribui um valor ao campo 'altura' da primeira posição do array.

p1 = pessoas[0]; // A variável p1 recebe os valores da primeira posição do array 'pessoas'.

if (p1.idade > 20) {

p1.idade++; // Incrementa a idade de p1 em 1, caso seja maior que 20.

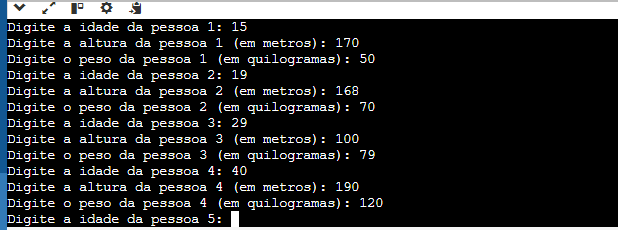
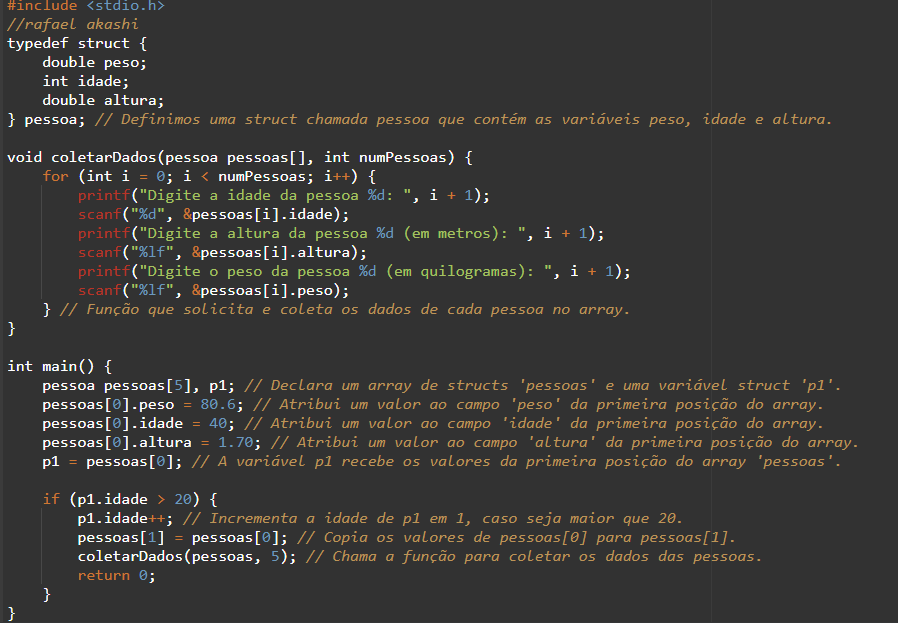
pessoas[1] = pessoas[0]; // Copia os valores de pessoas[0] para pessoas[1].

coletarDados(pessoas, 5); // Chama a função para coletar os dados das pessoas.

return 0;

}

}



**EXERCICIO 2**

#include <stdio.h> // Biblioteca para entrada e saída

#include <string.h> // Biblioteca para manipular strings

#include <locale.h> // Biblioteca para definir localização

// rafael akashi

typedef struct {

char modelo[50]; // Armazena o modelo do carro

int ano; // Armazena o ano do carro

double preco; // Armazena o preço do carro

} Carro; // Declara a estrutura Carro com modelo, ano e preço.

void modificarPreco(Carro \*c, double novoPreco) {

c->preco = novoPreco; // Atualiza o campo 'preco' do carro para o novo valor especificado.

}

void imprimirCarro(Carro \*c) {

printf("Modelo: %s\n", c->modelo); // Exibe o modelo do carro

printf("Ano: %d\n", c->ano); // Exibe o ano do carro

printf("Preço: R$ %.2lf\n", c->preco); // Exibe o preço do carro, formatado com duas casas decimais

}

int main() {

setlocale(LC\_ALL, "portuguese"); // Define a localização para português

Carro carro1; // Declara uma variável do tipo Carro chamada carro1

strcpy(carro1.modelo, "Xyz"); // Atribui o valor "Xyz" ao campo modelo de carro1

carro1.ano = 2020; // Define o ano de carro1 como 2020

carro1.preco = 90000.00; // Define o preço de carro1 como 90000.00

printf("Dados antes da modificação:\n");

imprimirCarro(&carro1); // Exibe os dados de carro1 antes da modificação

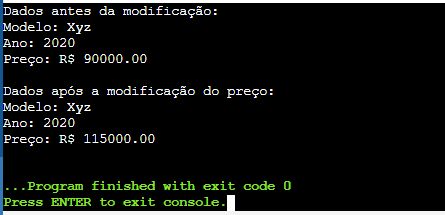
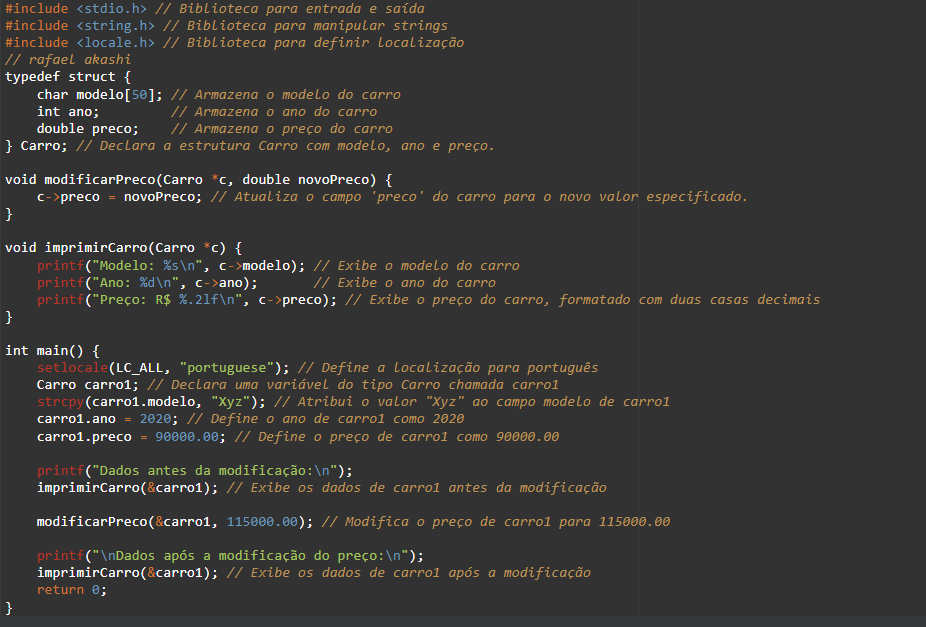
modificarPreco(&carro1, 115000.00); // Modifica o preço de carro1 para 115000.00

printf("\nDados após a modificação do preço:\n");

imprimirCarro(&carro1); // Exibe os dados de carro1 após a modificação

return 0;

}

****

**EXERCICIO 3**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

int calcularResultado(int v[], int n); // Declara a função calcularResultado.

int main() {

int vetor[5], i, resultado; // Declara o vetor, uma variável i e uma variável para armazenar o resultado.

for (i = 0; i < 5; i++) { // Loop para preencher o vetor com valores.

vetor[i] = i \* 2; // Atribui a cada posição do vetor o valor de i multiplicado por 2.

}

resultado = calcularResultado(vetor, 5); // Chama a função calcularResultado passando o vetor e seu tamanho.

printf("O resultado é: %d\n", resultado); // Imprime o resultado da soma dos elementos do vetor.

return 0; // Fim do programa.

}

int calcularResultado(int v[], int n) {

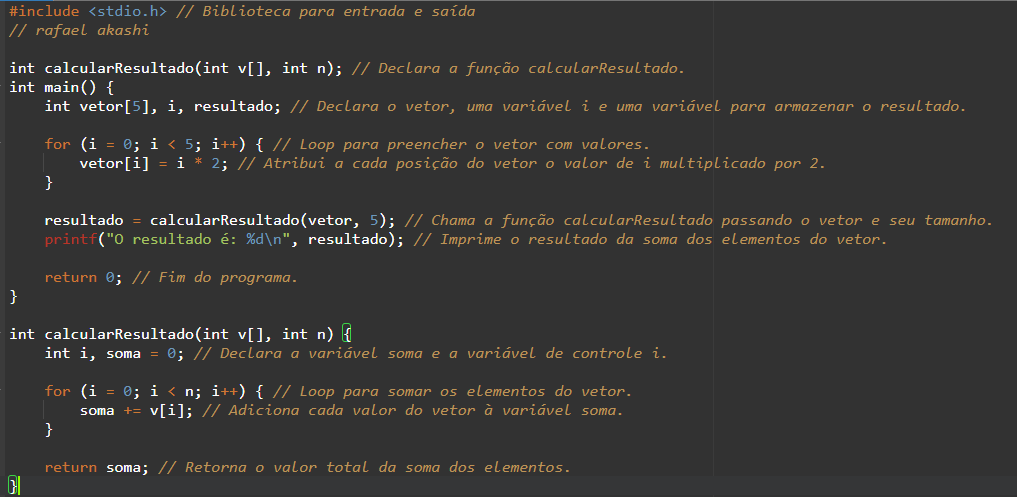
int i, soma = 0; // Declara a variável soma e a variável de controle i.

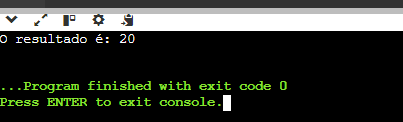
for (i = 0; i < n; i++) { // Loop para somar os elementos do vetor.

soma += v[i]; // Adiciona cada valor do vetor à variável soma.

}

return soma; // Retorna o valor total da soma dos elementos.

}



**EXERCICIO 4**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

float calcularMedia(float nota1, float nota2, float nota3, float nota4) {

return (nota1 + nota2 + nota3 + nota4) / 4.0; // Calcula a média das quatro notas.

}

int main() {

float nota1, nota2, nota3, nota4, media; // Declara variáveis para as notas e a média.

printf("Digite a primeira nota: ");

scanf("%f", &nota1); // Lê a primeira nota.

printf("Digite a segunda nota: ");

scanf("%f", &nota2); // Lê a segunda nota.

printf("Digite a terceira nota: ");

scanf("%f", &nota3); // Lê a terceira nota.

printf("Digite a quarta nota: ");

scanf("%f", &nota4); // Lê a quarta nota.

media = calcularMedia(nota1, nota2, nota3, nota4); // Calcula a média das notas usando a função calcularMedia.

printf("A média é: %.2f\n", media); // Exibe a média calculada.

if (media >= 7.0) { // Verifica se a média é suficiente para aprovação.

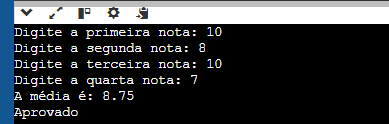
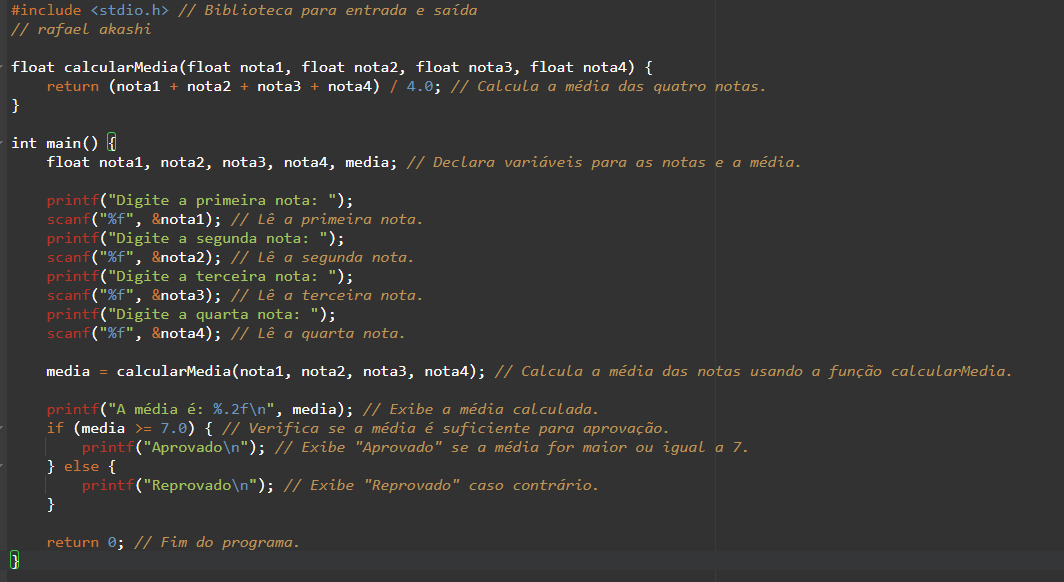
printf("Aprovado\n"); // Exibe "Aprovado" se a média for maior ou igual a 7.

} else {

printf("Reprovado\n"); // Exibe "Reprovado" caso contrário.

}

return 0; // Fim do programa.

}

**EXERCICIO 5**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

void impares(int limite) {

printf("Números ímpares de 1 até %d (exceto múltiplos de 7):\n", limite);

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

if (i % 2 != 0 && i % 7 != 0) { // Verifica se o número é ímpar e não é múltiplo de 7.

printf("%d ", i); // Imprime o número que atende à condição.

}

}

printf("\n");

}

int main() {

int numero; // Declara uma variável para armazenar o número limite.

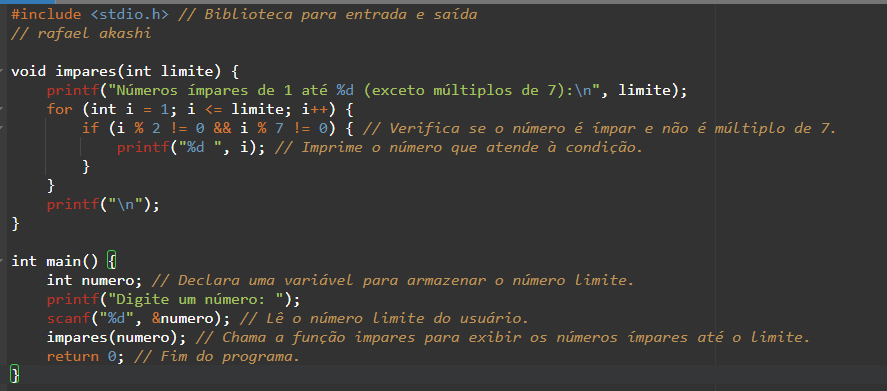
printf("Digite um número: ");

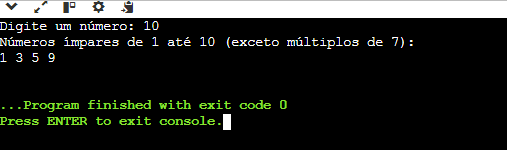
scanf("%d", &numero); // Lê o número limite do usuário.

impares(numero); // Chama a função impares para exibir os números ímpares até o limite.

return 0; // Fim do programa.

}





**EXERCICIO 6**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

float somarValores(int vetor[], int tamanho) {

float soma = 0.0; // Inicializa a variável soma com zero.

for (int i = 0; i < tamanho; i++) { // Percorre cada elemento do vetor.

soma += vetor[i]; // Adiciona o valor do elemento atual à soma.

}

return soma; // Retorna a soma total dos elementos do vetor.

}

int main() {

int tamanho; // Declara a variável para armazenar o tamanho do vetor.

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

scanf("%d", &tamanho); // Lê o tamanho do vetor inserido pelo usuário.

int vetor[tamanho]; // Declara o vetor com o tamanho especificado.

for (int i = 0; i < tamanho; i++) { // Preenche o vetor com valores inseridos pelo usuário.

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

scanf("%d", &vetor[i]); // Armazena o valor no vetor.

}

float soma = somarValores(vetor, tamanho); // Calcula a soma dos valores do vetor.

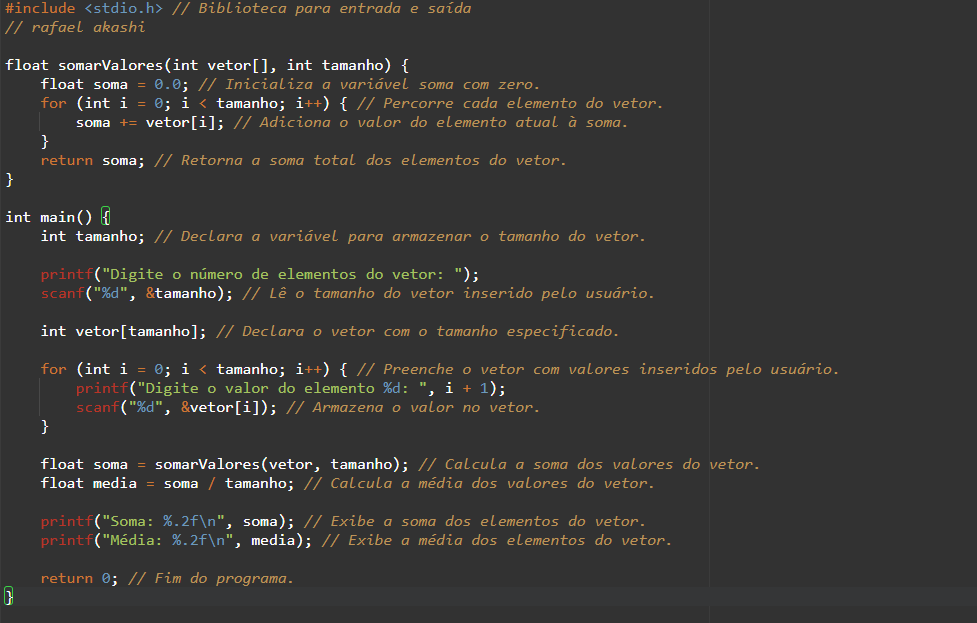
float media = soma / tamanho; // Calcula a média dos valores do vetor.

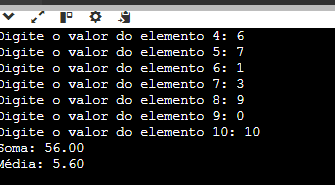
printf("Soma: %.2f\n", soma); // Exibe a soma dos elementos do vetor.

printf("Média: %.2f\n", media); // Exibe a média dos elementos do vetor.

return 0; // Fim do programa.

}





**EXERCICIO 7**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

void maiorValor(int vetor[], int tamanho, int \*maior, int \*posicao) {

\*maior = vetor[0]; // Inicializa o maior valor com o primeiro elemento do vetor.

\*posicao = 0; // Inicializa a posição do maior valor como a posição 0.

for (int i = 1; i < tamanho; i++) { // Percorre o vetor a partir do segundo elemento.

if (vetor[i] > \*maior) { // Verifica se o elemento atual é maior que o maior atual.

\*maior = vetor[i]; // Atualiza o maior valor.

\*posicao = i; // Atualiza a posição do maior valor.

}

}

}

int main() {

int tamanho; // Declara a variável para o tamanho do vetor.

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

scanf("%d", &tamanho); // Lê o tamanho do vetor inserido pelo usuário.

int vetor[tamanho]; // Declara o vetor com o tamanho especificado.

for (int i = 0; i < tamanho; i++) { // Lê os valores dos elementos do vetor.

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

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

}

int maior, posicao; // Declara variáveis para armazenar o maior valor e sua posição.

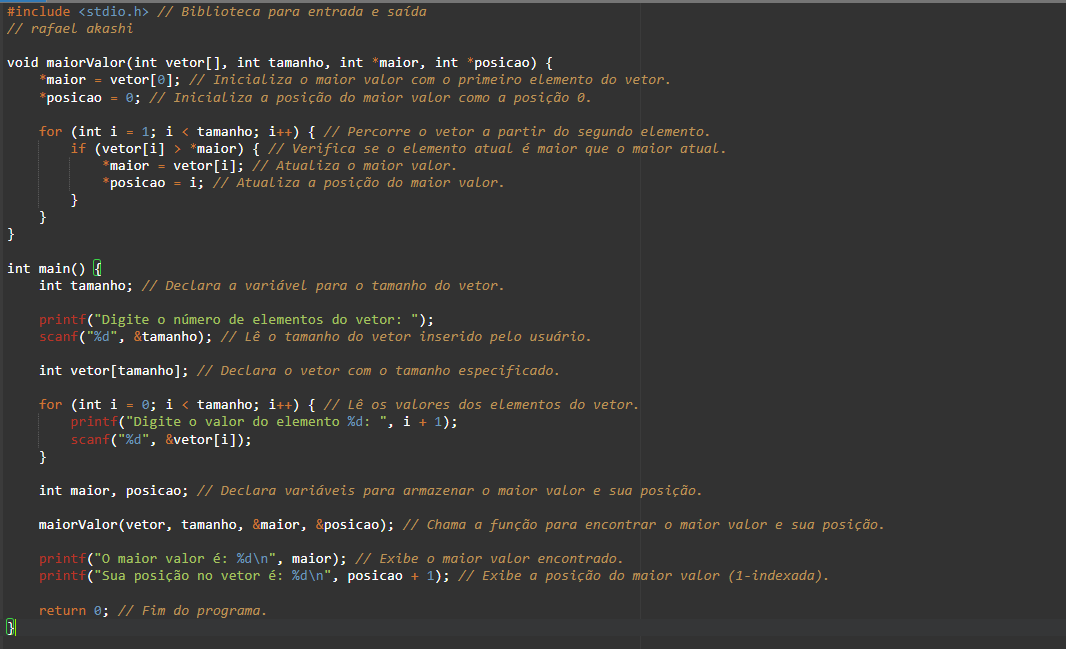
maiorValor(vetor, tamanho, &maior, &posicao); // Chama a função para encontrar o maior valor e sua posição.

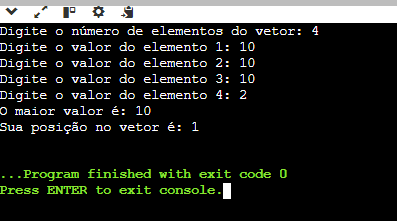
printf("O maior valor é: %d\n", maior); // Exibe o maior valor encontrado.

printf("Sua posição no vetor é: %d\n", posicao + 1); // Exibe a posição do maior valor (1-indexada).

return 0; // Fim do programa.

}





**EXERCICIO 8**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

void pares(int numero) {

printf("Números pares de 1 até %d:\n", numero);

for (int i = 1; i <= numero; i++) { // Percorre os números de 1 até o número especificado.

if (i % 2 == 0) { // Verifica se o número é par.

printf("%d ", i); // Imprime o número par.

}

}

printf("\n");

}

int main() {

int numero; // Declara uma variável para armazenar o número inserido pelo usuário.

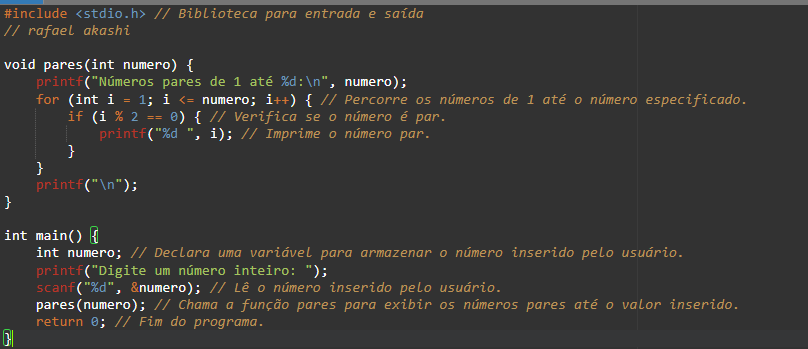
printf("Digite um número inteiro: ");

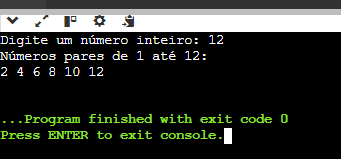
scanf("%d", &numero); // Lê o número inserido pelo usuário.

pares(numero); // Chama a função pares para exibir os números pares até o valor inserido.

return 0; // Fim do programa.

}





**EXERCICIO 9**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

float multiplicarValores(int vetor[], int tamanho) {

float produto = 1.0; // Inicializa o produto como 1.0.

for (int i = 0; i < tamanho; i++) { // Percorre cada elemento do vetor.

produto \*= vetor[i]; // Multiplica o valor atual do vetor ao produto total.

}

return produto; // Retorna o produto total dos elementos do vetor.

}

int main() {

int tamanho; // Declara a variável para o tamanho do vetor.

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

scanf("%d", &tamanho); // Lê o tamanho do vetor inserido pelo usuário.

int vetor[tamanho]; // Declara o vetor com o tamanho especificado.

for (int i = 0; i < tamanho; i++) { // Lê os valores dos elementos do vetor.

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

scanf("%d", &vetor[i]); // Armazena o valor no vetor.

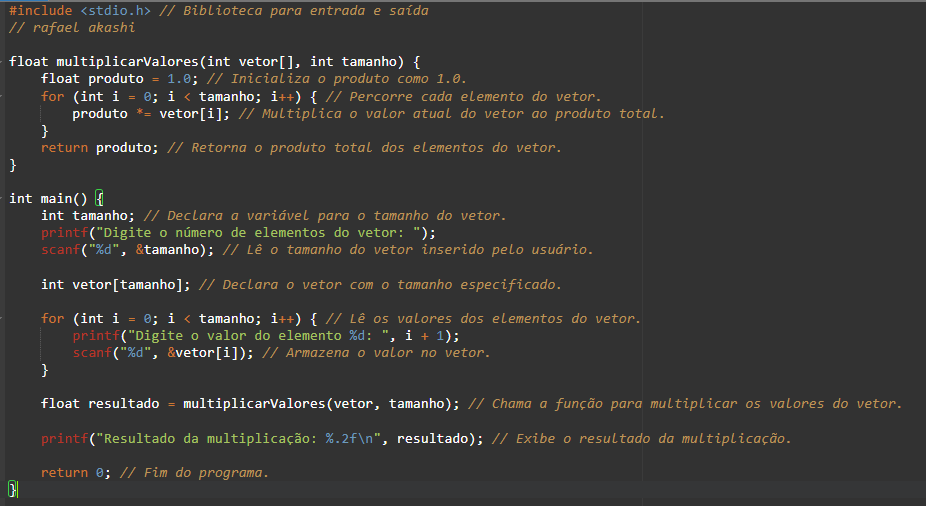
}

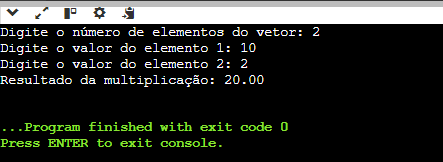
float resultado = multiplicarValores(vetor, tamanho); // Chama a função para multiplicar os valores do vetor.

printf("Resultado da multiplicação: %.2f\n", resultado); // Exibe o resultado da multiplicação.

return 0; // Fim do programa.

}





**EXERCICIO 10**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

void maiorNumero(int num1, int num2) {

if (num1 > num2) { // Verifica se o primeiro número é maior que o segundo.

printf("O maior número é: %d\n", num1); // Imprime o maior número.

} else if (num2 > num1) { // Verifica se o segundo número é maior que o primeiro.

printf("O maior número é: %d\n", num2); // Imprime o maior número.

} else { // Caso em que os números são iguais.

printf("Os números são iguais: %d\n", num1); // Informa que os números são iguais.

}

}

int main() {

int numero1, numero2; // Declara variáveis para armazenar os números.

printf("Digite o primeiro número inteiro: ");

scanf("%d", &numero1); // Lê o primeiro número.

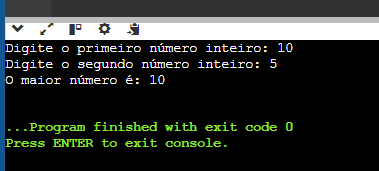
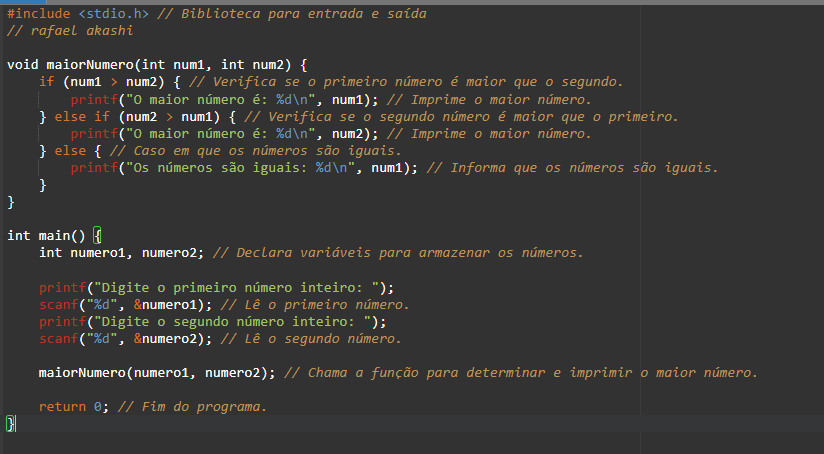
printf("Digite o segundo número inteiro: ");

scanf("%d", &numero2); // Lê o segundo número.

maiorNumero(numero1, numero2); // Chama a função para determinar e imprimir o maior número.

return 0; // Fim do programa.

}



**EXERCICIO 11**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

int menorNumero(int n1, int n2, int n3) {

int menor = n1; // Inicializa a variável menor com o primeiro número.

if (n2 < menor) { // Compara o segundo número com o menor atual.

menor = n2; // Atualiza o menor se n2 for menor.

}

if (n3 < menor) { // Compara o terceiro número com o menor atual.

menor = n3; // Atualiza o menor se n3 for menor.

}

return menor; // Retorna o menor número encontrado.

}

int main() {

int numero1, numero2, numero3; // Declara variáveis para armazenar os números.

printf("Digite o primeiro número inteiro: ");

scanf("%d", &numero1); // Lê o primeiro número.

printf("Digite o segundo número inteiro: ");

scanf("%d", &numero2); // Lê o segundo número.

printf("Digite o terceiro número inteiro: ");

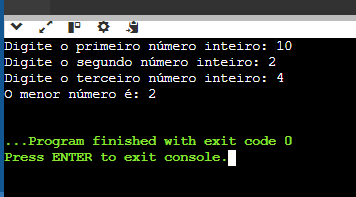
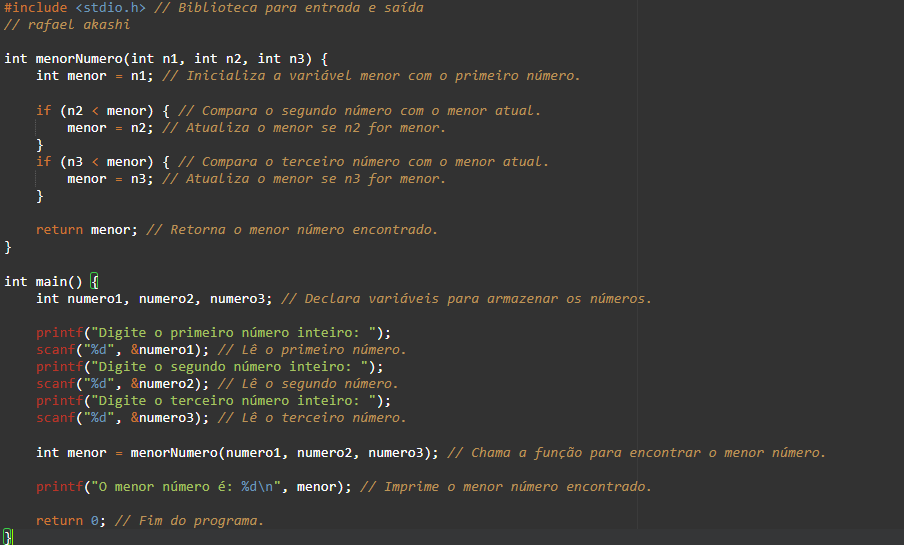
scanf("%d", &numero3); // Lê o terceiro número.

int menor = menorNumero(numero1, numero2, numero3); // Chama a função para encontrar o menor número.

printf("O menor número é: %d\n", menor); // Imprime o menor número encontrado.

return 0; // Fim do programa.

}



**EXERCICIO 12**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

void inverso(int vetor[], int tamanho) {

printf("Números na ordem inversa:\n");

for (int i = tamanho - 1; i >= 0; i--) { // Percorre o vetor de trás para frente.

printf("%d ", vetor[i]); // Imprime cada número na ordem inversa.

}

printf("\n"); // Nova linha após a impressão.

}

int main() {

int vetor[10]; // Declara um vetor para armazenar 10 números inteiros.

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

for (int i = 0; i < 10; i++) { // Loop para ler os 10 números do usuário.

printf("Número %d: ", i + 1); // Solicita o número ao usuário.

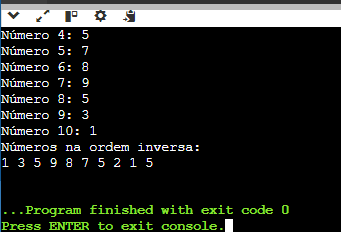
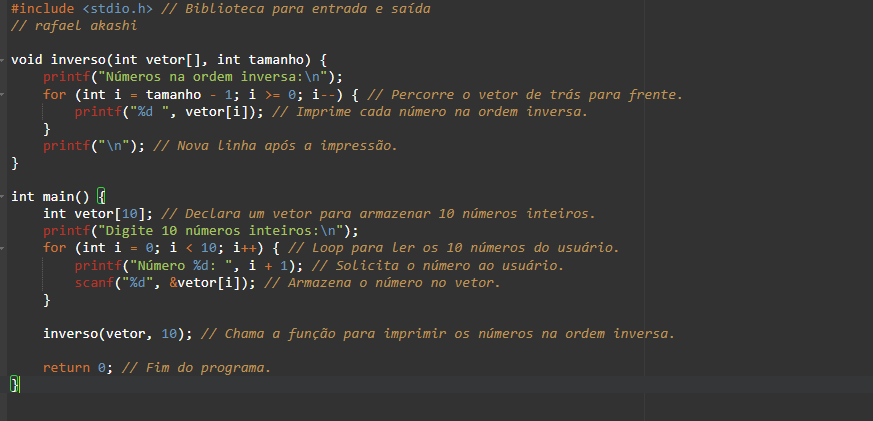
scanf("%d", &vetor[i]); // Armazena o número no vetor.

}

inverso(vetor, 10); // Chama a função para imprimir os números na ordem inversa.

return 0; // Fim do programa.

}



**EXERCICIO 13**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

float calcularMedia(int num1, int num2, int num3) {

return (num1 + num2 + num3) / 3.0; // Calcula e retorna a média dos três números.

}

int main() {

int numero1, numero2, numero3; // Declara variáveis para armazenar os três números.

printf("Digite o primeiro número inteiro: ");

scanf("%d", &numero1); // Lê o primeiro número inteiro.

printf("Digite o segundo número inteiro: ");

scanf("%d", &numero2); // Lê o segundo número inteiro.

printf("Digite o terceiro número inteiro: ");

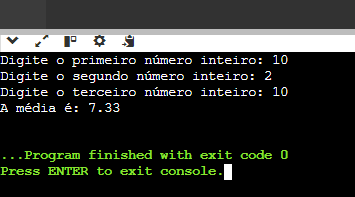
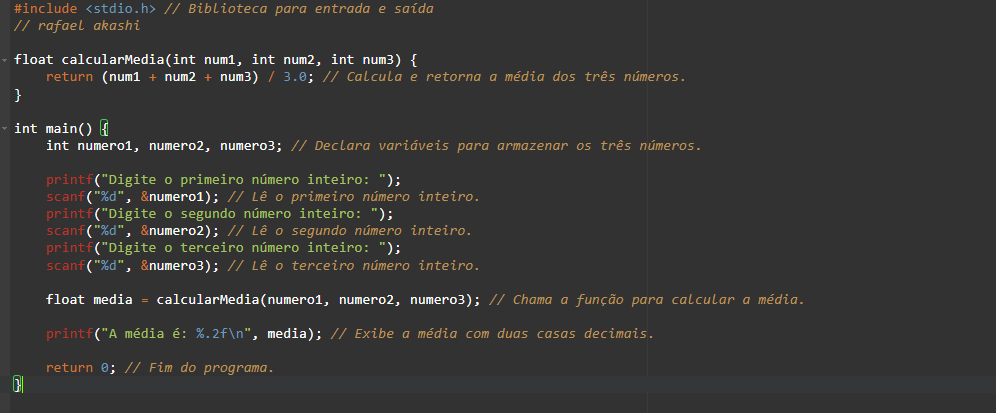
scanf("%d", &numero3); // Lê o terceiro número inteiro.

float media = calcularMedia(numero1, numero2, numero3); // Chama a função para calcular a média.

printf("A média é: %.2f\n", media); // Exibe a média com duas casas decimais.

return 0; // Fim do programa.

}



**EXERCICIO 14**

#include <stdio.h> // Biblioteca para entrada e saída

// rafael akashi

int calcularResultado(int soma, int primeiroValor, int ultimoValor) {

return soma + (primeiroValor \* 5) + (ultimoValor \* 2); // Calcula e retorna o resultado com base na soma e valores extremos do vetor.

}

int main() {

int vetor[10]; // Declara um vetor de 10 inteiros.

int soma = 0; // Inicializa a soma como zero.

// Preenche o vetor com números de 1 a 10 e calcula a soma.

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

vetor[i] = i + 1; // Atribui a cada posição do vetor o valor de i + 1 (1 a 10).

soma += vetor[i]; // Adiciona o valor atual à soma total.

}

int primeiroValor = vetor[0]; // O primeiro valor do vetor.

int ultimoValor = vetor[9]; // O último valor do vetor.

int resultado = calcularResultado(soma, primeiroValor, ultimoValor); // Chama a função para calcular o resultado.

printf("O resultado é: %d\n", resultado); // Exibe o resultado final.

return 0; // Fim do programa.

}

