1. **Crie uma função que receba um valor e diga se é par ou ímpar.**

#include <stdio.h>

void parOuImpar(int num) {

if (num % 2 == 0) {

printf("%d é par.\n", num);

} else {

printf("%d é ímpar.\n", num);

}

}

int main() {

int num;

printf("Insira um número: ");

scanf("%d", &num);

parOuImpar(num);

return 0;

}

1. **Crie uma função que receba um valor e informe se ele é positivo ou não.**

#include <stdio.h>

void verificaPositivo(int valor) {

if(valor >= 0) {

printf("%d é um número positivo.\n", valor);

} else {

printf("%d é um número negativo.\n", valor);

}

}

int main() {

int numero;

printf("Digite um número: ");

scanf("%d", &numero);

verificaPositivo(numero);

return 0;

}

1. **Crie uma função que receba três valores, 'a', 'b' e 'c', que são os coeficientes de uma equação do segundo grau e retorne o valor do delta, que é dado por 'b² - 4ac'**

#include <stdio.h>

#include <math.h>

float calcularDelta(float a, float b, float c) {

float delta = pow(b, 2) - 4 \* a \* c;

return delta;

}

int main() {

float a, b, c, delta;

printf("Insira o valor de a: ");

scanf("%f", &a);

printf("Insira o valor de b: ");

scanf("%f", &b);

printf("Insira o valor de c: ");

scanf("%f", &c);

delta = calcularDelta(a, b, c);

printf("O delta é: %f\n", delta);

return 0;

}

1. **Usando as 3 funções acima, crie um aplicativo que calcula as raízes de uma equação do 2o grau:**

**ax² + bx + c=0**

**Para ela existir, o coeficiente 'a' deve ser diferente de zero.**

**Caso o delta seja maior ou igual a zero, as raízes serão reais. Caso o delta seja negativo, as reais serão complexas e da forma: x + iy.**

#include <stdio.h>

#include <math.h>

void calculate\_delta(float a, float b, float c, float \*delta) {

\*delta = (b \* b) - (4 \* a \* c);

}

void calculate\_real\_roots(float a, float b, float delta, float \*x1, float \*x2) {

\*x1 = (-b + sqrt(delta)) / (2 \* a);

\*x2 = (-b - sqrt(delta)) / (2 \* a);

}

void calculate\_complex\_roots(float a, float b, float delta, float \*x1\_real, float \*x1\_imag, float \*x2\_real, float \*x2\_imag) {

float real\_part = -b / (2 \* a);

float imaginary\_part = sqrt(-delta) / (2 \* a);

\*x1\_real = real\_part;

\*x1\_imag = imaginary\_part;

\*x2\_real = real\_part;

\*x2\_imag = -imaginary\_part;

}

int main() {

float a, b, c, delta, x1, x2, x1\_real, x1\_imag, x2\_real, x2\_imag;

printf("Insira o coeficiente a: ");

scanf("%f", &a);

if (a == 0) {

printf("O coeficiente a não pode ser igual a zero.\n");

return 1;

}

printf("Insira o coeficiente b: ");

scanf("%f", &b);

printf("Insira o coeficiente c: ");

scanf("%f", &c);

calculate\_delta(a, b, c, &delta);

if (delta >= 0) {

calculate\_real\_roots(a, b, delta, &x1, &x2);

printf("As raizes reais são: %.2f e %.2f\n", x1, x2);

} else {

calculate\_complex\_roots(a, b, delta, &x1\_real, &x1\_imag, &x2\_real, &x2\_imag);

printf("As raizes complexas são: %.2f + %.2fi e %.2f - %.2fi\n", x1\_real, x1\_imag, x2\_real, x2\_imag);

}

return 0;

}

1. **Crie uma função em linguagem C que receba 2 números e retorne o maior valor.**

include <stdio.h>

int encontrarMaior(int num1, int num2) {

if (num1 > num2) {

return num1;

} else {

return num2;

}

}

int main() {

int num1, num2, maior;

printf("Insira o primeiro número: ");

scanf("%d", &num1);

printf("Insira o segundo número: ");

scanf("%d", &num2);

maior = encontrarMaior(num1, num2);

printf("O maior número é: %d\n", maior);

return 0;

}

1. **Crie uma função em linguagem C que receba 2 números e retorne o menor valor.**

#include <stdio.h>

int encontrarMenor(int num1, int num2) {

if (num1 < num2) {

return num1;

} else {

return num2;

}

}

int main() {

int num1, num2, menor;

printf("Insira o primeiro número: ");

scanf("%d", &num1);

printf("Insira o segundo número: ");

scanf("%d", &num2);

menor = encontrarMenor(num1, num2);

printf("O menor número é: %d\n", menor);

return 0;

}

1. **Crie uma função em linguagem C que receba 3 números e retorne o maior valor, use a função da questão 4.**

#include <stdio.h>

void find\_min\_max(int arr[], int size, int \*min, int \*max) {

\*min = arr[0];

\*max = arr[0];

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

if (arr[i] < \*min) {

\*min = arr[i];

}

if (arr[i] > \*max) {

\*max = arr[i];

}

}

}

int main() {

int arr[] = {3, 5, 2, 8, 1, 9};

int size = sizeof(arr) / sizeof(arr[0]);

int min, max;

find\_min\_max(arr, size, &min, &max);

printf("O menor valor é: %d\n", min);

printf("O maior valor é: %d\n", max);

return 0;

}

1. **Crie uma função em linguagem C que receba 3 números e retorne o menor valor, use a função da questão 5.**

#include <stdio.h>

int menor(int a, int b){

if(a < b)

return a;

else

return b;

}

int menor\_de\_tres(int a, int b, int c){

return menor(menor(a, b), c);

}

int main(){

int num1, num2, num3;

printf("Digite 3 números: ");

scanf("%d %d %d", &num1, &num2, &num3);

printf("O menor valor é: %d", menor\_de\_tres(num1, num2, num3));

return 0;

}6#include <stdio.h>

void find\_min\_max(int arr[], int size, int \*min, int \*max) {

\*min = arr[0];

\*max = arr[0];

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

if (arr[i] < \*min) {

\*min = arr[i];

}

if (arr[i] > \*max) {

\*max = arr[i];

}

}

}

int main() {

int arr[] = {3, 5, 2, 8, 1, 9};

int size = sizeof(arr) / sizeof(arr[0]);

int min, max;

find\_min\_max(arr, size, &min, &max);

printf("O menor valor é: %d\n", min);

printf("O maior valor é: %d\n", max);

return 0;

}

1. **Crie uma função em linguagem C chamado Dado() que retorna, através de sorteio, um número de 1 até 6. Utilize as funções srand(time(NULL)) e rand();**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int Dado() {

srand(time(NULL)); // inicializa o gerador de números aleatórios com o valor atual do relógio

return (rand() % 6) + 1; // retorna um número aleatório entre 1 e 6

}

int main() {

int resultado = Dado(); // chama a função Dado() para gerar um número aleatório

printf("O resultado do dado foi: %d\n", resultado);

return 0;

}

1. **Use a função da questão passado e lance o dado 1 milhão de vezes. Conte quantas vezes cada número saiu. A probabilidade deu certo? Ou seja, a porcentagem dos números foi parecida?**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int Dado() {

return (rand() % 6) + 1;

}

int main() {

int n = 1000000;

int freq[7] = {0}; // inicializa todas as frequências com zero

srand(time(NULL)); // inicializa a semente do gerador de números aleatórios com o tempo atual

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

int num = Dado();

freq[num]++;

}

printf("Frequencia de cada numero:\n");

for (int i = 1; i <= 6; i++) {

float porcentagem = ((float) freq[i] / n) \* 100;

printf("%d: %d vezes (%.2f%%)\n", i, freq[i], porcentagem);

}

return 0;

}

1. **Crie um aplicativo de conversão entre as temperaturas Celsius e Farenheit.**

**Primeiro o usuário deve escolher se vai entrar com a temperatura em Célsius ou Farenheit, depois a conversão escolhida é realizada através de um comando SWITCH.**

**Se C é a temperatura em Célsius e F em farenheit, as fórmulas de conversão são:**

**C= 5.(F-32)/9**

**F= (9.C/5) + 32**

#include <stdio.h>

int main() {

float temperatura, resultado;

int opcao;

printf("Escolha a opcao de conversao:\n");

printf("1 - Celsius para Farenheit\n");

printf("2 - Farenheit para Celsius\n");

scanf("%d", &opcao);

switch(opcao) {

case 1:

printf("Digite a temperatura em Celsius: ");

scanf("%f", &temperatura);

resultado = (9 \* temperatura / 5) + 32;

printf("%.2f C = %.2f F\n", temperatura, resultado);

break;

case 2:

printf("Digite a temperatura em Farenheit: ");

scanf("%f", &temperatura);

resultado = 5 \* (temperatura - 32) / 9;

printf("%.2f F = %.2f C\n", temperatura, resultado);

break;

default:

printf("Opcao invalida!\n");

break;

}

return 0;

}