**Functions:**

#include <iostream>

using namespace std;

**// Function declaration (prototype)**

int add(int a, int b);

int subtract(int a, int b);

int multiply(int a, int b);

float divide(int a, int b);

int main() {

int num1, num2;

// Input from the user

cout << "Enter first number: ";

cin >> num1;

cout << "Enter second number: ";

cin >> num2;

// Calling functions and displaying the results

cout << "Addition: " << add(num1, num2) << endl;

cout << "Subtraction: " << subtract(num1, num2) << endl;

cout << "Multiplication: " << multiply(num1, num2) << endl;

cout << "Division: " << divide(num1, num2) << endl;

return 0;

}

// Function definitions

int add(int a, int b) {

return a + b;

}

int subtract(int a, int b) {

return a - b;

}

int multiply(int a, int b) {

return a \* b;

}

float divide(int a, int b) {

if (b != 0) {

return static\_cast<float>(a) / b; // Casting to float for division

} else {

cout << "Error: Division by zero is not allowed." << endl;

return 0;

}

}

#include <iostream>

using namespace std;

**// Function to calculate factorial using recursion**

int factorial(int n) {

if (n == 0 || n == 1) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

int main() {

int num;

cout << "Enter a number to find its factorial: ";

cin >> num;

if (num < 0) {

cout << "Factorial is not defined for negative numbers." << endl;

} else {

cout << "Factorial of " << num << " is " << factorial(num) << endl;

}

return 0;

}

#include <iostream>

using namespace std;

**// Function to check whether a number is prime or not**

bool isPrime(int num) {

if (num <= 1) {

return false; // 0 and 1 are not prime numbers

}

for (int i = 2; i \* i <= num; ++i) { // Check divisibility up to sqrt(num)

if (num % i == 0) {

return false; // If divisible, it's not prime

}

}

return true;

}

int main() {

int num;

cout << "Enter a number to check if it is prime: ";

cin >> num;

if (isPrime(num)) {

cout << num << " is a prime number." << endl;

} else {

cout << num << " is not a prime number." << endl;

}

return 0;

}

#include <iostream>

using namespace std;

**// Function to swap two numbers**

void swapNumbers(int &a, int &b) {

int temp = a;

a = b;

b = temp;

}

int main() {

int num1, num2;

cout << "Enter first number: ";

cin >> num1;

cout << "Enter second number: ";

cin >> num2;

cout << "Before swapping: " << "num1 = " << num1 << ", num2 = " << num2 << endl;

// Calling function to swap the numbers

swapNumbers(num1, num2);

cout << "After swapping: " << "num1 = " << num1 << ", num2 = " << num2 << endl;

return 0;

}

#include <iostream>

using namespace std;

**// Function declarations**

int add(int a, int b);

int subtract(int a, int b);

int multiply(int a, int b);

float divide(int a, int b);

int main() {

int num1, num2, choice;

cout << "Enter first number: ";

cin >> num1;

cout << "Enter second number: ";

cin >> num2;

cout << "Choose an operation:" << endl;

cout << "1. Add" << endl;

cout << "2. Subtract" << endl;

cout << "3. Multiply" << endl;

cout << "4. Divide" << endl;

cin >> choice;

switch (choice) {

case 1:

cout << "Result: " << add(num1, num2) << endl;

break;

case 2:

cout << "Result: " << subtract(num1, num2) << endl;

break;

case 3:

cout << "Result: " << multiply(num1, num2) << endl;

break;

case 4:

if (num2 != 0) {

cout << "Result: " << divide(num1, num2) << endl;

} else {

cout << "Error: Division by zero is not allowed." << endl;

}

break;

default:

cout << "Invalid choice." << endl;

break;

}

return 0;

}

// Function definitions

int add(int a, int b) {

return a + b;

}

int subtract(int a, int b) {

return a - b;

}

int multiply(int a, int b) {

return a \* b;

}

float divide(int a, int b) {

return static\_cast<float>(a) / b;

}

**Pointer :**

#include <iostream>

using namespace std;

int main() {

int num = 10;

int \*ptr; // Pointer declaration

ptr = &num; // Storing the address of num in ptr

cout << "Value of num: " << num << endl;

cout << "Address of num: " << &num << endl; // Address of num

cout << "Pointer ptr points to address: " << ptr << endl;

cout << "Value at the address pointed by ptr: " << \*ptr << endl; // Dereferencing the pointer

return 0;

}

#include <iostream>

using namespace std;

**// Function to swap values using pointers**

void swapNumbers(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int num1, num2;

cout << "Enter first number: ";

cin >> num1;

cout << "Enter second number: ";

cin >> num2;

cout << "Before swapping: num1 = " << num1 << ", num2 = " << num2 << endl;

// Calling the swap function

swapNumbers(&num1, &num2);

cout << "After swapping: num1 = " << num1 << ", num2 = " << num2 << endl;

return 0;

}

#include <iostream>

using namespace std;

int main() {

int arr[5] = {1, 2, 3, 4, 5};

int \*ptr = arr; // Pointer to the first element of the array

**// Accessing elements of the array using the pointer**

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

cout << "Element " << i + 1 << " is: " << \*(ptr + i) << endl;

}

return 0;

}

**String :**

**Length of string:**

#include <iostream>

#include <string>

using namespace std;

int main() {

string str;

cout << "Enter a string: ";

cin >> str; // Input a single word (no spaces)

cout << "Length of the string: " << str.length() << endl;

return 0;

}

#include <iostream>

#include <string>

using namespace std;

int main() {

string str1, str2;

cout << "Enter first string: ";

cin >> str1;

cout << "Enter second string: ";

cin >> str2;

string result = str1 + " " + str2; // **Concatenate the strings**

cout << "Concatenated string: " << result << endl;

return 0;

}

**Comparison string:**

#include <iostream>

#include <string>

using namespace std;

int main() {

string str1, str2;

cout << "Enter first string: ";

cin >> str1;

cout << "Enter second string: ";

cin >> str2;

if (str1 == str2) {

cout << "Strings are equal." << endl;

} else {

cout << "Strings are not equal." << endl;

}

return 0;

}

**Reverse string:**

#include <iostream>

#include <string>

using namespace std;

int main() {

string str;

cout << "Enter a string: ";

cin >> str;

string reversed = string(str.rbegin(), str.rend()); // Reverse the string using iterators

cout << "Reversed string: " << reversed << endl;

return 0;

}

**Find substring:**

#include <iostream>

#include <string>

using namespace std;

int main() {

string str, substring;

cout << "Enter a string: ";

cin >> str;

cout << "Enter substring to find: ";

cin >> substring;

if (str.find(substring) != string::npos) {

cout << "Substring found!" << endl;

} else {

cout << "Substring not found." << endl;

}

return 0;

}

**Number of voval in string:**

#include <iostream>

#include <string>

using namespace std;

int main() {

string str;

int count = 0;

cout << "Enter a string: ";

getline(cin, str);

for (char c : str) {

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||

c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') {

count++;

}

}

cout << "Number of vowels: " << count << endl;

return 0;

}

Palandrom :

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

int main() {

string str;

cout << "Enter a string: ";

getline(cin, str);

string original = str;

reverse(str.begin(), str.end()); // reverse the string

if (original == str) {

cout << "The string is a palindrome." << endl;

} else {

cout << "The string is not a palindrome." << endl;

}

return 0;

}