1)Write a function t check if a given string is a palindrome or not

#include<iostream>

using namespace std;

bool isPalindrome(string str) {

int start = 0;

int end = str.length() - 1;

while (start < end) {

if (str[start] != str[end]) {

return false;

}

start++;

end--;

}

return true;

}

int main() {

string str;

cout << "Enter a string: ";

cin >> str;

if (isPalindrome(str)) {

cout << str << " is a palindrome." << endl;

} else {

cout << str << " is not a palindrome." << endl;

}

return 0;

}

2) Write a function to calculate the area of a circle given its radius.

#include <iostream>

#include <cmath>

double calculateCircleArea(double radius) {

const double PI = M\_PI;

return PI \* radius \* radius;

}

int main() {

double radius;

std::cout << "Enter the radius of the circle: ";

std::cin >> radius;

if (radius < 0) {

std::cout << "Radius cannot be negative." << std::endl;

return 1;

}

double area = calculateCircleArea(radius);

std::cout << "The area of the circle with radius " << radius << " is " << area << "." << std::endl;

return 0;

}

3) Write a function to convert a given temperature from Celsius to Fahrenheit

#include <iostream>

double celsiusToFahrenheit(double celsius) {

return (celsius \* 9.0 / 5.0) + 32.0;

}

int main() {

double celsius;

std::cout << "Enter temperature in Celsius: ";

std::cin >> celsius;

double fahrenheit = celsiusToFahrenheit(celsius);

std::cout << celsius << " degrees Celsius is equal to " << fahrenheit << " degrees Fahrenheit." << std::endl;

return 0;

}

4) Write a function to count the number of words in a given string

#include <iostream>

#include <string>

int countWords(const std::string& str) {

int wordCount = 0;

bool inWord = false;

for (char ch : str) {

if (isspace(ch)) {

inWord = false;

} else if (!inWord) {

inWord = true;

++wordCount;

}

}

return wordCount;

}

int main() {

std::string input;

std::cout << "Enter a string: ";

std::getline(std::cin, input);

int wordCount = countWords(input);

std::cout << "Number of words: " << wordCount << std::endl;

return 0;

}

5) Write a function to find the minimum and maximum elements in a given array.

#include <iostream>

#include <climits>

void findMinMax(const int arr[], int size, int& min, int& max) {

min = INT\_MAX;

max = INT\_MIN;

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

if (arr[i] < min) {

min = arr[i];

}

if (arr[i] > max) {

max = arr[i];

}

}

}

int main() {

int arr[] = {10, 5, 8, 20, 3, 15};

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

int min, max;

findMinMax(arr, size, min, max);

std::cout << "Minimum element: " << min << std::endl;

std::cout << "Maximum element: " << max << std::endl;

return 0;

}

6) Write a function to reverse a given string.

#include <iostream>

#include <string>

void reverseString(std::string& str) {

int n = str.length();

for (int i = 0; i < n / 2; ++i) {

std::swap(str[i], str[n - i - 1]);

}

}

int main() {

std::string input;

std::cout << "Enter a string: ";

std::getline(std::cin, input);

reverseString(input);

std::cout << "Reversed string: " << input << std::endl;

return 0;

}

7) Write a function to determine if a given integer is a prime number or not.

#include <iostream>

#include <cmath>

bool isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i <= std::sqrt(num); ++i) {

if (num % i == 0) {

return false;

}

}

return true;

}

int main() {

int num;

std::cout << "Enter a number: ";

std::cin >> num;

if (isPrime(num)) {

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

} else {

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

}

return 0;

}

8) Write a function to determine the GCD (greatest common divisor) of two given integers.

#include <iostream>

int gcd(int a, int b) {

if (b == 0) {

return a;

}

return gcd(b, a % b);

}

int main() {

int num1, num2;

std::cout << "Enter two integers: ";

std::cin >> num1 >> num2;

int result = gcd(num1, num2);

std::cout << "The GCD of " << num1 << " and " << num2 << " is: " << result << std::endl;

return 0;

}

9) Write a function to compute the factorial of a given integer

#include <iostream>

int factorial(int n) {

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

return 1;

}

return n \* factorial(n - 1);

}

int main() {

int num;

std::cout << "Enter a positive integer: ";

std::cin >> num;

if (num < 0) {

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

} else {

int result = factorial(num);

std::cout << "The factorial of " << num << " is: " << result << std::endl;

}

return 0;

}

10) Write a C++ program to add 2 matrics

#include<iostream>

using namespace std;

int main()

{

int mat1[2][2], mat2[2][2], i, j, mat3[2][2];

cout<<"Enter Elements of First Matrix: ";

for(i=0; i<2; i++)

{

for(j=0; j<2; j++)

cin>>mat1[i][j];

}

cout<<"Enter Elements of Second Matrix: ";

for(i=0; i<2; i++)

{

for(j=0; j<2; j++)

cin>>mat2[i][j];

}

cout<<"\nAdding the Two Given Matrix...\n";

for(i=0; i<2; i++)

{

for(j=0; j<2; j++)

mat3[i][j] = mat1[i][j]+mat2[i][j];

}

cout<<"Addition Result of Two Given Matrix is:\n";

for(i=0; i<2; i++)

{

for(j=0; j<2; j++)

cout<<mat3[i][j]<<" ";

cout<<endl;

}

return 0;

}

11) Write a C++ program to find the number of occurrences of a value in an array

#include <iostream>

int main() {

int size, targetValue;

std::cout << "Enter the number of elements in the array: ";

std::cin >> size;

if (size <= 0) {

std::cerr << "The size of the array must be positive." << std::endl;

return 1;

}

const int MAX\_SIZE = 100;

if (size > MAX\_SIZE) {

std::cerr << "Size exceeds the maximum allowed size of " << MAX\_SIZE << "." << std::endl;

return 1;

}

int array[MAX\_SIZE];

std::cout << "Enter " << size << " integer values:" << std::endl;

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

std::cin >> array[i];

}

std::cout << "Enter the value to find occurrences of: ";

std::cin >> targetValue;

int count = 0;

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

if (array[i] == targetValue) {

++count;

}

}

std::cout << "The value " << targetValue << " occurs " << count << " times in the array." << std::endl;

return 0;

}

12) Write a C++ program to find the average of elements in an array

#include <iostream>

#include <iomanip>

int main() {

const int MAX\_SIZE = 100;

int size;

std::cout << "Enter the number of elements in the array (up to " << MAX\_SIZE << "): ";

std::cin >> size;

if (size <= 0 || size > MAX\_SIZE) {

std::cerr << "Array size must be between 1 and " << MAX\_SIZE << "." << std::endl;

return 1;

}

int array[MAX\_SIZE];

std::cout << "Enter " << size << " integer values:" << std::endl;

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

std::cin >> array[i];

}

int sum = 0;

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

sum += array[i];

}

double average = static\_cast<double>(sum) / size;

std::cout << "The average of the elements is " << std::fixed << std::setprecision(2) << average << "." << std::endl;

return 0;

}

13) Write a C++ program to reverse the elements of an array.

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int size;

cout << "Enter the number of elements in the array: ";

cin >> size;

if (size <= 0) {

cout << "Array size should be a positive integer." << endl;

return 1;

}

vector<int> arr(size);

cout << "Enter the elements of the array:" << endl;

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

cin >> arr[i];

}

cout << "Original array:" << endl;

for (const int& num : arr) {

cout << num << " ";

}

cout << endl;

reverse(arr.begin(), arr.end());

cout << "Reversed array:" << endl;

for (const int& num : arr) {

cout << num << " ";

}

cout << endl;

return 0;

}

14) Write a program to read in two integers and perform the following operations on them: addition, subtraction, multiplication, division, and modulo.

#include <iostream>

using namespace std;

int main()

{

int first, second, add, subtract, multiply,modulo;

float divide;

cout << "Please enter two integer: ";

cin >> first;

cin >> second;

add = first + second;

subtract = first - second;

multiply = first \* second;

divide = first / (float)second;

modulo = first % second;

cout << endl <<"Sum = " << add;

cout << endl <<"Difference = " << subtract;

cout << endl <<"Multiplication = " << multiply;

cout << endl <<"Division = " << divide;

cout << endl <<"Modulo = " << modulo;

return 0;

}

15) Write a program to read in a character and determine if it is a vowel or a consonant.

#include<iostream>

#include<cctype>

int main(){

char ch;

std::cin>>ch;

if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U'){

std::cout<<ch<<"is a vowel";

}else{

std::cout<<ch<<"is a consonant";

}

return 0;

}

16) Write a program to read in an integer and determine if it is positive, negative, or zero.

#include<iostream>

int main()

{

int num;

std::cout<<"enter a number:";

std::cin>>num;

if(num>0){

std::cout<<num<<"is a positive number";

}else if(num<0){

std::cout<<num<<"is a negative number";

}else{

std::cout<<num<<"is zero";

}

return 0;

}

17) Write a program to read in an integer and determine if it is even or odd.

#include<iostream>

int main()

{

int num;

std::cout<<"enter a number";

std::cin>>num;

if(num%2==0){

std::cout<<num<< "is even";

}else{

std::cout<<num<< "is odd";

}

return 0;

}

18) Write a program to read in an integer and print the multiplication table for that number using a for loop.

#include <iostream>

int main() {

int number;

std::cout << "Enter an integer: ";

std::cin >> number;

std::cout << "Multiplication table for " << number << ":" << std::endl;

for (int i = 1; i <= 10; ++i) {

std::cout << number << " x " << i << " = " << (number \* i) << std::endl;

}

return 0;

}

19) Write a C++ program to merge two arrays into a single array.

#include <iostream>

#include <vector>

int main() {

int size1, size2;

std::cout << "Enter the number of elements in the first array: ";

std::cin >> size1;

std::cout << "Enter the number of elements in the second array: ";

std::cin >> size2;

std::vector<int> vector1(size1);

std::vector<int> vector2(size2);

std::cout << "Enter " << size1 << " elements for the first array:" << std::endl;

for (int i = 0; i < size1; ++i) {

std::cin >> vector1[i];

}

// Input elements for the second vector

std::cout << "Enter " << size2 << " elements for the second array:" << std::endl;

for (int i = 0; i < size2; ++i) {

std::cin >> vector2[i];

}

// Create a vector to hold the merged result

std::vector<int> mergedVector;

mergedVector.reserve(size1 + size2); // Reserve memory to avoid multiple allocations

// Merge the two vectors

mergedVector.insert(mergedVector.end(), vector1.begin(), vector1.end());

mergedVector.insert(mergedVector.end(), vector2.begin(), vector2.end());

// Display the merged vector

std::cout << "The merged array is:" << std::endl;

for (int value : mergedVector) {

std::cout << value << " ";

}

std::cout << std::endl;

return 0;

}

20) Write a C++ program to delete an element from an array at a specific position.

#include <iostream>

void deleteElementAtPosition(int arr[], int &size, int position) {

if (position < 0 || position >= size) {

std::cout << "Invalid position!" << std::endl;

return;

}

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

arr[i] = arr[i + 1];

}

--size;

}

void printArray(int arr[], int size) {

std::cout << "Array elements: ";

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

std::cout << arr[i] << " ";

}

std::cout << std::endl;

}

int main() {

const int MAX\_SIZE = 100;

int arr[MAX\_SIZE];

int size = 0;

std::cout << "Enter the number of elements: ";

std::cin >> size;

if (size > MAX\_SIZE || size <= 0) {

std::cout << "Invalid array size!" << std::endl;

return 1;

}

std::cout << "Enter " << size << " elements: ";

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

std::cin >> arr[i];

}

printArray(arr, size);

int position;

std::cout << "Enter the position of the element to delete (0-based index): ";

std::cin >> position;

deleteElementAtPosition(arr, size, position);

printArray(arr, size);

return 0;

}

21)