**DAY 1 LAB**

**EASY**

**1.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int var = 0 ;

const int cons = 15;

cout << "Initial Value:" ;

cout << "var: " << var ;

cout << "cons: " << cons ;

cout << "Final Value:" ;

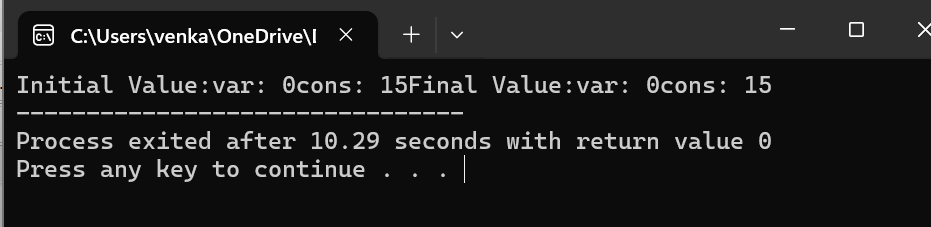
cout << "var: " << var ;

cout << "cons: " << cons ;

return 0;

}

**OUTPUT**



**2.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int integerNumber = 10;

cout << "Integer number: " << integerNumber <<endl;

float floatNumber = 3.14;

cout << "Float number: " << floatNumber <<endl;

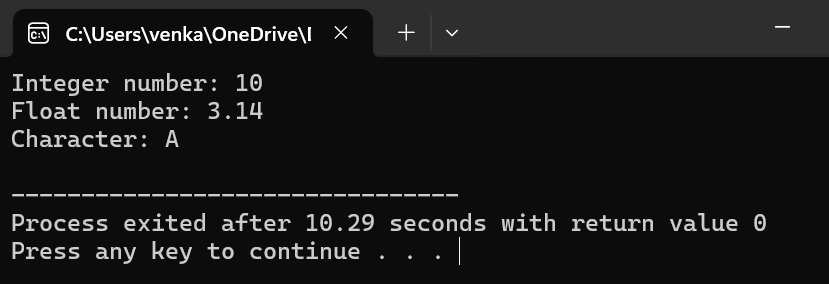
char character = 'A';

cout << "Character: " << character <<endl;

return 0;

}

**OUTPUT**



**3.PROGRAM**

#include <iostream>

int main()

{

int myInteger = 10;

float myFloat;

myFloat = static\_cast<float>(myInteger);

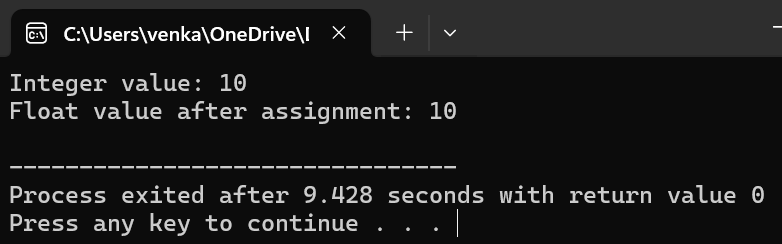
std::cout << "Integer value: " << myInteger << std::endl;

std::cout << "Float value after assignment: " << myFloat << std::endl;

return 0;

}

**OUTPUT**



**4.PROGRAM**

#include <iostream>

int main()

{

int myInteger = 23;

float myFloat;

myFloat = myInteger;

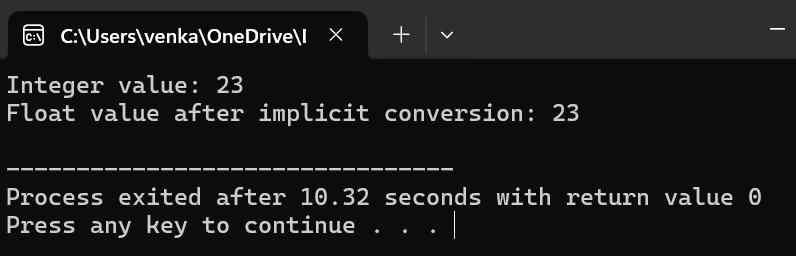
std::cout << "Integer value: " << myInteger << std::endl;

std::cout << "Float value after implicit conversion: " << myFloat << std::endl;

return 0;

}

**OUTPUT**



**5.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int num1 = 5;

int num2 = 7;

int sum = num1 + num2;

std::cout << "First integer: " << num1 << std::endl;

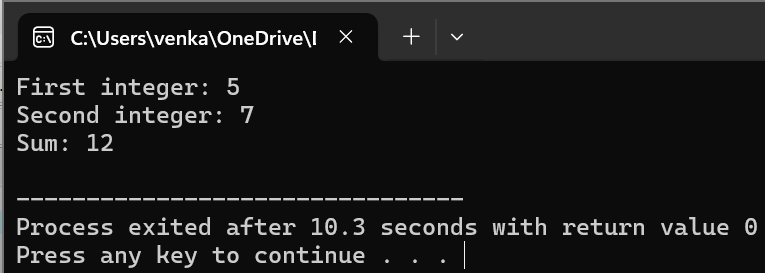
std::cout << "Second integer: " << num2 << std::endl;

std::cout << "Sum: " << sum << std::endl;

return 0;

}

**OUTPUT**



**6.PROGRAM**

#include <iostream>

int main()

{

int number;

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

std::cin >> number;

if (number % 2 == 0)

{

std::cout << number << " is even." << std::endl;

} else {

std::cout << number << " is odd." << std::endl;

}

return 0;

}

**OUTPUT**



**7.PROGRAM**

#include <iostream>

int main()

{

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

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

std::cout << "Elements of the array:" << std::endl;

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

{

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

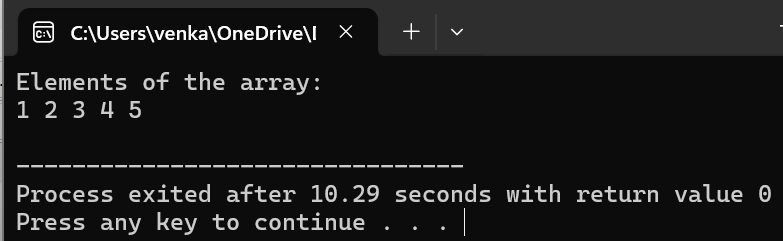
}

std::cout << std::endl;

return 0;

}

**OUTPUT**



**8.PROGRAM**

#include <iostream>

int main()

{

int dayNumber;

std::cout << "Enter a number (1-7) representing the day of the week: ";

std::cin >> dayNumber;

switch (dayNumber)

{

case 1:

std::cout << "Sunday" << std::endl;

break;

case 2:

std::cout << "Monday" << std::endl;

break;

case 3:

std::cout << "Tuesday" << std::endl;

break;

case 4:

std::cout << "Wednesday" << std::endl;

break;

case 5:

std::cout << "Thursday" << std::endl;

break;

case 6:

std::cout << "Friday" << std::endl;

break;

case 7:

std::cout << "Saturday" << std::endl;

break;

default:

std::cout << "Invalid input! Please enter a number between 1 and 7." << std::endl;

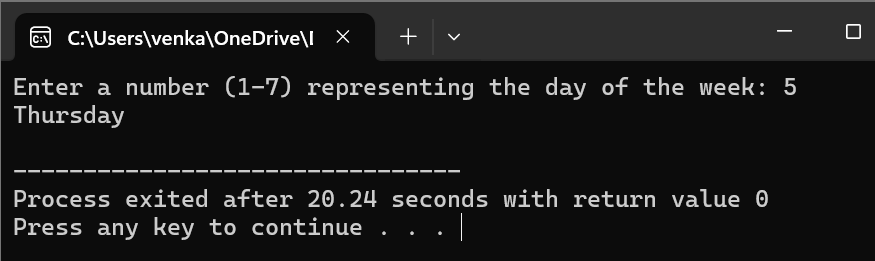
break;

}

return 0;

}

**OUTPUT**



**9.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

double length, width, area;

std::cout << "Enter the length of the rectangle: ";

std::cin >> length;

std::cout << "Enter the width of the rectangle: ";

std::cin >> width;

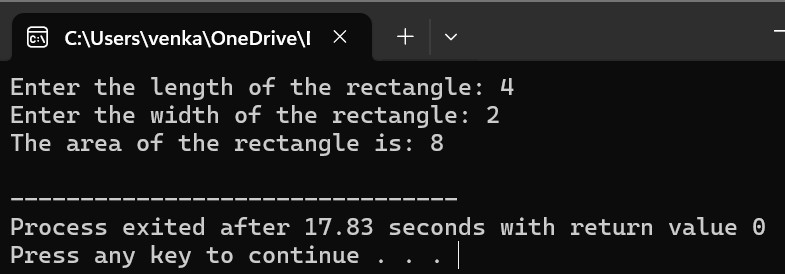
area = length \* width;

std::cout << "The area of the rectangle is: " << area << std::endl;

return 0;

}

**OUTPUT**



**10.PROGRAM**

#include <iostream>

int main()

{

double num1, num2, num3, max;

std::cout << "Enter three numbers: ";

std::cin >> num1 >> num2 >> num3;

if (num1 >= num2 && num1 >= num3)

{

max = num1;

} else if (num2 >= num1 && num2 >= num3)

{

max = num2;

} else {

max = num3;

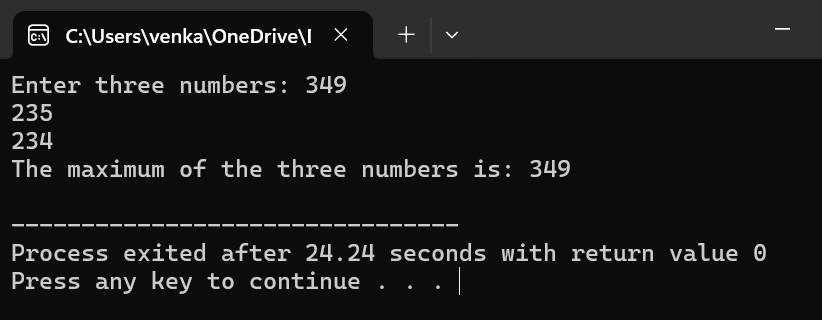
}

std::cout << "The maximum of the three numbers is: " << max << std::endl;

return 0;

}

**OUTPUT**



**MEDIUM**

**1.PROGRAM**

#include <iostream>

int main()

{

const double PI = 3.14159;

double radius, area;

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

std::cin >> radius;

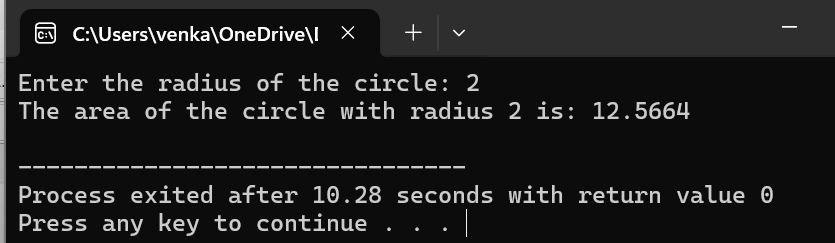
area = PI \* radius \* radius;

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

return 0;

}

**OUTPUT**



**2.PROGRAM**

#include <iostream>

using namespace std;

int main() {

int integerNum;

float floatNum;

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

std::cin >> integerNum;

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

std::cin >> floatNum;

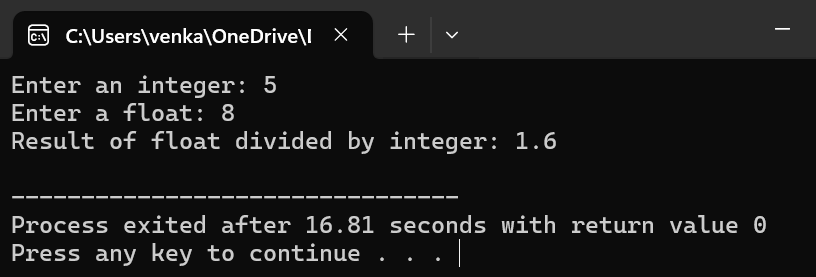
float result = floatNum / integerNum;

std::cout << "Result of float divided by integer: " << result << std::endl;

return 0;

}

**OUTPUT**



**3.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int year;

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

std::cin >> year;

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

std::cout << year << " is a leap year." << std::endl;

} else {

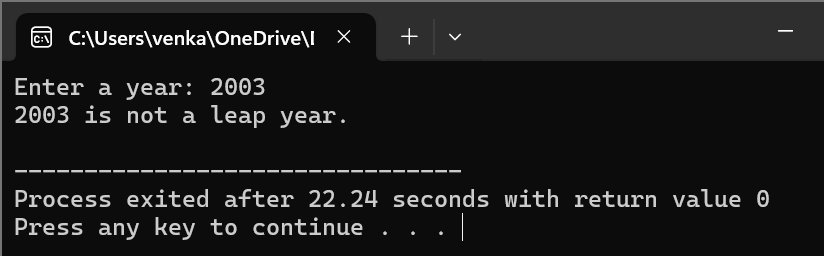
std::cout << year << " is not a leap year." << std::endl;

}

return 0;

}

**OUTPUT**



**4.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

float length, width, area;

cout << "Enter the length of the rectangle: ";

cin >> length;

cout << "Enter the width of the rectangle: ";

cin >> width;

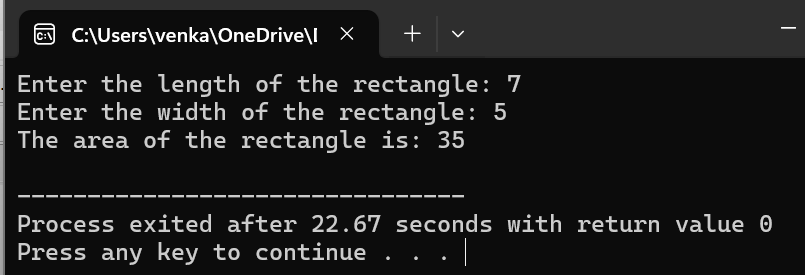
area = length \* width;

cout << "The area of the rectangle is: " << area << endl;

return 0;

}

**OUTPUT**



**5.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int number;

cout << "Enter an integer: ";

cin >> number;

if (number & 1)

{

cout << number << " is an odd number." << endl;

} else {

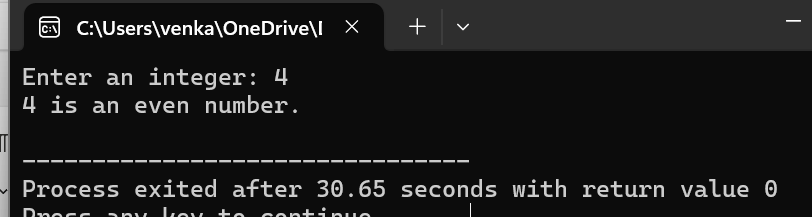
cout << number << " is an even number." << endl;

}

return 0;

}

**OUTPUT**



**6.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int monthNumber;

cout << "Enter a month number (1-12): ";

cin >> monthNumber;

switch (monthNumber)

{

case 1:

cout << "January" << endl;

break;

case 2:

cout << "February" << endl;

break;

case 3:

cout << "March" << endl;

break;

case 4:

cout << "April" << endl;

break;

case 5:

cout << "May" << endl;

break;

case 6:

cout << "June" << endl;

break;

case 7:

cout << "July" << endl;

break;

case 8:

cout << "August" << endl;

break;

case 9:

cout << "September" << endl;

break;

case 10:

cout << "October" << endl;

break;

case 11:

cout << "November" << endl;

break;

case 12:

cout << "December" << endl;

break;

default:

cout << "Invalid month number. Please enter a number between 1 and 12." << endl;

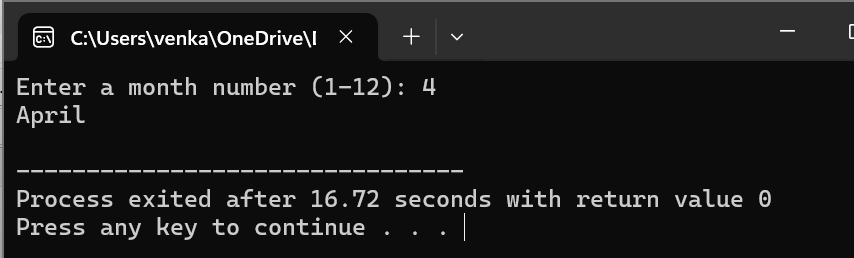
break;

}

return 0;

}

**OUTPUT**



**7.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

const float PI = 3.14159;

float radius, volume;

cout << "Enter the radius of the sphere: ";

cin >> radius;

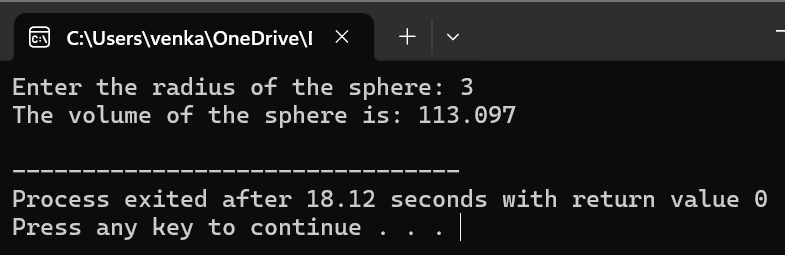
volume = (4.0 / 3.0) \* PI \* radius \* radius \* radius;

cout << "The volume of the sphere is: " << volume << endl;

return 0;

}

**OUTPUT**



**8.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int dividend, divisor;

float result;

cout << "Enter the dividend: ";

cin >> dividend;

cout << "Enter the divisor: ";

cin >> divisor;

if (divisor == 0)

{

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

} else {

result = static\_cast<float>(dividend) / divisor;

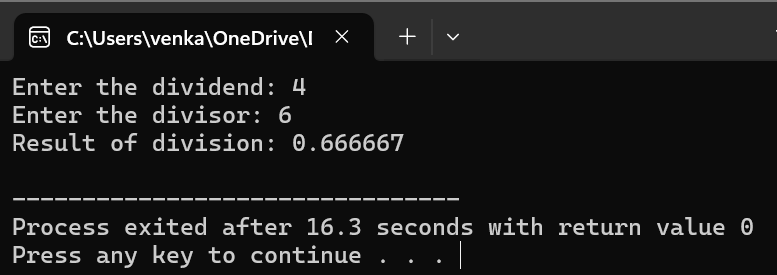
cout << "Result of division: " << result << endl;

}

return 0;

}

**OUTPUT**



**9.PROGRAM**

#include <iostream>

using namespace std;

class Complex

{

private:

float real;

float imag;

public:

Complex(float r = 0.0, float i = 0.0) : real(r), imag(i)

{}

Complex operator+(const Complex& other) const

{

return Complex(real + other.real, imag + other.imag);

}

Complex operator-(const Complex& other) const

{

return Complex(real - other.real, imag - other.imag);

}

void display() const

{

cout << "(" << real << " + " << imag << "i)" << endl;

}

};

int main()

{

Complex num1(3.0, 4.0);

Complex num2(1.0, 2.0);

Complex sum = num1 + num2;

cout << "Sum: ";

sum.display();

Complex diff = num1 - num2;

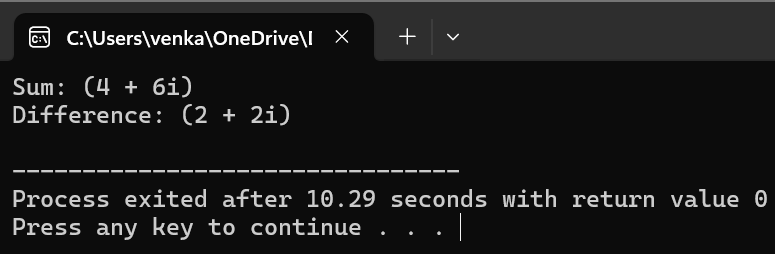
cout << "Difference: ";

diff.display();

return 0;

}

**OUTPUT**



**10.PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int number;

unsigned long long factorial = 1;

cout << "Enter a number: ";

cin >> number;

for (int i = 1; i <= number; ++i)

{

factorial \*= i;

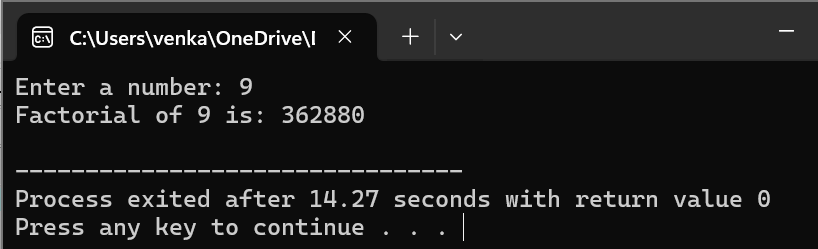
}

cout << "Factorial of " << number << " is: " << factorial << endl;

return 0;

}

**OUTPUT**



**HARD**

**1.PROGRAM**

#include <iostream>

using namespace std;

int fibonacci(int n)

{

if (n <= 1)

return n;

else

return fibonacci(n - 1) + fibonacci(n - 2);

}

void displayFibonacci(int n)

{

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

{

cout << fibonacci(i) << " ";

}

}

int main()

{

int n;

cout << "Enter the number of terms: ";

cin >> n;

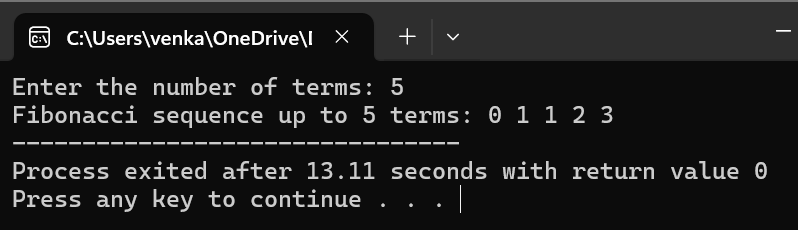
cout << "Fibonacci sequence up to " << n << " terms: ";

displayFibonacci(n);

return 0;

}

OUTPUT



2.PROGRAM

#include <iostream>

using namespace std;

int main()

{

int size;

cout << "Enter the size of the array: ";

cin >> size;

int\* arr = new (nothrow) int[size];

if (arr == nullptr)

{

cout << "Memory allocation failed!";

return 1;

}

cout << "Enter " << size << " integers:" << endl;

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

{

cin >> arr[i];

}

cout << "Values stored in the array:" << endl;

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

{

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

}

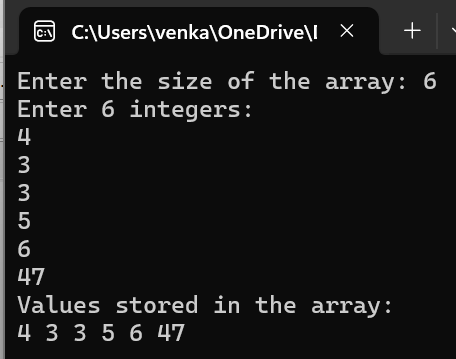
cout << endl;

delete[] arr;

return 0;

}

OUTPUT



3.PROGRAM

#include <iostream>

#include <vector>

using namespace std;

class Matrix

{

private:

vector<vector<int>> data;

int rows;

int cols;

public:

Matrix(int numRows, int numCols) : rows(numRows), cols(numCols)

{

data.resize(rows, vector<int>(cols, 0));

}

Matrix operator+(const Matrix& other) const

{

if (rows != other.rows || cols != other.cols)

{

cerr << "Error: Matrices must have the same dimensions for addition." << endl;

return \*this;

}

Matrix result(rows, cols);

for (int i = 0; i < rows; ++i)

{

for (int j = 0; j < cols; ++j)

{

result.data[i][j] = data[i][j] + other.data[i][j];

}

}

return result;

}

Matrix operator-(const Matrix& other) const

{

if (rows != other.rows || cols != other.cols)

{

cerr << "Error: Matrices must have the same dimensions for subtraction." << endl;

return \*this;

}

Matrix result(rows, cols);

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

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

result.data[i][j] = data[i][j] - other.data[i][j];

}

}

return result;

}

Matrix operator\*(const Matrix& other) const

{

if (cols != other.rows) {

cerr << "Error: Number of columns in the first matrix must match the number of rows in the second matrix for multiplication." << endl;

return \*this;

}

Matrix result(rows, other.cols);

for (int i = 0; i < rows; ++i)

{

for (int j = 0; j < other.cols; ++j)

{

for (int k = 0; k < cols; ++k)

{

result.data[i][j] += data[i][k] \* other.data[k][j];

}

}

}

return result;

}

void setValue(int row, int col, int value)

{

if (row >= 0 && row < rows && col >= 0 && col < cols)

{

data[row][col] = value;

} else {

cerr << "Error: Invalid row or column index." << endl;

}

}

void display() const

{

for (int i = 0; i < rows; ++i)

{

for (int j = 0; j < cols; ++j)

{

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

}

cout << endl;

}

}

};

int main()

{

Matrix mat1(2, 3);

mat1.setValue(0, 0, 1);

mat1.setValue(0, 1, 2);

mat1.setValue(0, 2, 3);

mat1.setValue(1, 0, 4);

mat1.setValue(1, 1, 5);

mat1.setValue(1, 2, 6);

Matrix mat2(2, 3);

mat2.setValue(0, 0, 7);

mat2.setValue(0, 1, 8);

mat2.setValue(0, 2, 9);

mat2.setValue(1, 0, 10);

mat2.setValue(1, 1, 11);

mat2.setValue(1, 2, 12);

cout << "Matrix 1:" << endl;

mat1.display();

cout << "Matrix 2:" << endl;

mat2.display();

cout << "Matrix Addition:" << endl;

(mat1 + mat2).display();

cout << "Matrix Subtraction:" << endl;

(mat1 - mat2).display();

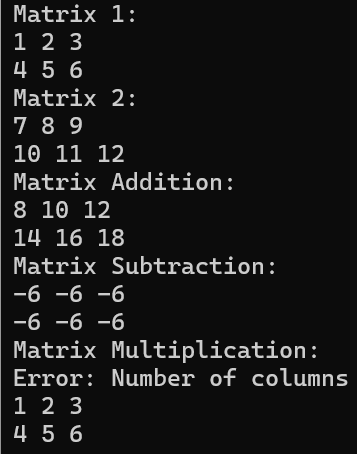
cout << "Matrix Multiplication:" << endl;

(mat1 \* mat2).display();

return 0;

}

OUTPUT



4.PROGRAM

#include<iostream>

using namespace std;

int area(int);

int area(int,int);

float area(float);

float area(float,float);

int main()

{

int s,l,b;

float r,bs,ht;

cout<<"Enter side of a square:";

cin>>s;

cout<<"Enter length and breadth of rectangle:";

cin>>l>>b;

cout<<"Enter radius of circle:";

cin>>r;

cout<<"Enter base and height of triangle:";

cin>>bs>>ht;

cout<<"Area of square is"<<area(s);

cout<<"\nArea of rectangle is "<<area(l,b);

cout<<"\nArea of circle is "<<area(r);

cout<<"\nArea of triangle is "<<area(bs,ht);

}

int area(int s)

{

return(s\*s);

}

int area(int l,int b)

{

return(l\*b);

}

float area(float r)

{

return(3.14\*r\*r);

}

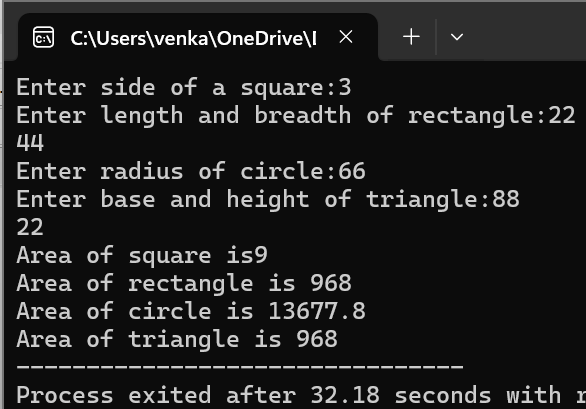
float area(float bs,float ht)

{

return((bs\*ht)/2);

}

OUTPUT



5.PROGRAM

#include <iostream>

#include <stack>

#include <string>

#include <cmath>

using namespace std;

bool isOperator(char c)

{

return (c == '+' || c == '-' || c == '\*' || c == '/');

}

int precedence(char op) {

if (op == '+' || op == '-')

return 1;

if (op == '\*' || op == '/')

return 2;

return 0;

}

double applyOperator(double a, double b, char op)

{

switch(op) {

case '+': return a + b;

case '-': return a - b;

case '\*': return a \* b;

case '/': return a / b;

default: return 0;

}

}

double evaluateExpression(const string& expression)

{

stack<double> values;

stack<char> operators;

for (int i = 0; i < expression.length(); ++i)

{

if (expression[i] == ' ')

continue;

if (isdigit(expression[i]))

{

string numStr;

while (i < expression.length() && (isdigit(expression[i]) || expression[i] == '.'))

{

numStr += expression[i];

++i;

}

--i;

values.push(stod(numStr));

}

else if (expression[i] == '(')

{

operators.push('(');

}

else if (expression[i] == ')')

{

while (!operators.empty() && operators.top() != '(')

{

double b = values.top();

values.pop();

double a = values.top();

values.pop();

char op = operators.top();

operators.pop();

values.push(applyOperator(a, b, op));

}

operators.pop();

}

else if (isOperator(expression[i]))

{

while (!operators.empty() && precedence(operators.top()) >= precedence(expression[i]))

{

double b = values.top();

values.pop();

double a = values.top();

values.pop();

char op = operators.top();

operators.pop();

values.push(applyOperator(a, b, op));

}

operators.push(expression[i]);

}

}

while (!operators.empty())

{

double b = values.top();

values.pop();

double a = values.top();

values.pop();

char op = operators.top();

operators.pop();

values.push(applyOperator(a, b, op));

}

return values.top();

}

int main()

{

string expression;

cout << "Enter a mathematical expression: ";

getline(cin, expression);

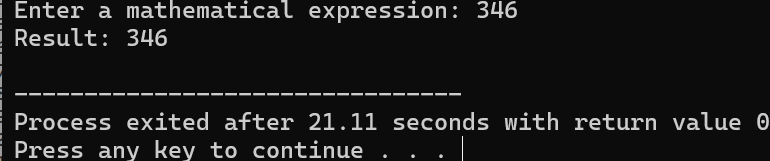
double result = evaluateExpression(expression);

cout << "Result: " << result << endl;

return 0;

}

OUTPUT



6.PROGRAM

#include <iostream>

using namespace std;

template<typename T>

T findMax(const T arr[], int size)

{

T maxElement = arr[0];

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

{

if (arr[i] > maxElement)

{

maxElement = arr[i];

}

}

return maxElement;

}

int main()

{

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

int intMax = findMax(intArray, 5);

cout << "Maximum element in the integer array: " << intMax << endl;

float floatArray[] = {3.5f, 7.2f, 1.8f, 6.4f, 9.1f};

float floatMax = findMax(floatArray, 5);

cout << "Maximum element in the float array: " << floatMax << endl;

double doubleArray[] = {2.3, 9.8, 5.6, 1.2, 7.4};

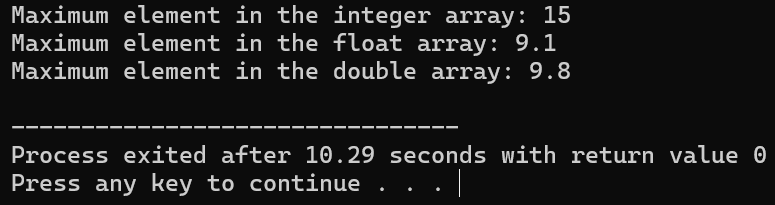
double doubleMax = findMax(doubleArray, 5);

cout << "Maximum element in the double array: " << doubleMax << endl;

return 0;

}

OUTPUT



7.PROGRAM

#include <iostream>

using namespace std;

void towerOfHanoi(int n, char source, char destination, char auxiliary)

{

if (n == 1)

{

cout << "Move disk 1 from rod " << source << " to rod " << destination << endl;

return;

}

towerOfHanoi(n - 1, source, auxiliary, destination);

cout << "Move disk " << n << " from rod " << source << " to rod " << destination << endl;

towerOfHanoi(n - 1, auxiliary, destination, source);

}

int main()

{

int numDisks;

cout << "Enter the number of disks: ";

cin >> numDisks;

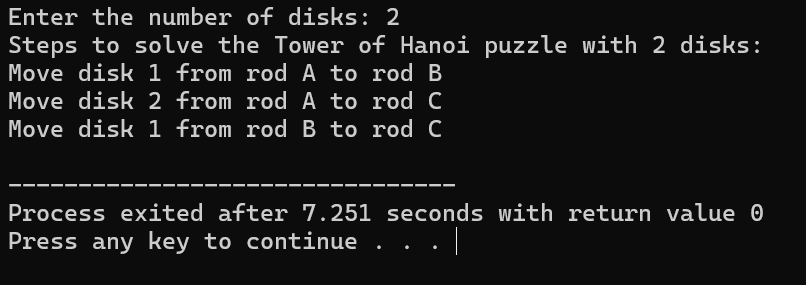
cout << "Steps to solve the Tower of Hanoi puzzle with " << numDisks << " disks:" << endl;

towerOfHanoi(numDisks, 'A', 'C', 'B');

return 0;

}

OUTPUT



8.PROGRAM

#include <iostream>

#include <fstream>

#include <vector>

#include <stdexcept>

using namespace std;

double calculateAverage(const vector<double>& data)

{

if (data.empty())

{

throw invalid\_argument("No data to calculate average.");

}

double sum = 0;

for (double num : data)

{

sum += num;

}

return sum / data.size();

}

int main()

{

string filename;

cout << "Enter the name of the file: ";

cin >> filename;

ifstream inputFile(filename);

if (!inputFile) {

cerr << "Error: Unable to open file." << endl;

return 1;

}

vector<double> data;

double value;

while (inputFile >> value)

{

data.push\_back(value);

}

if (inputFile.eof())

{

try {

double average = calculateAverage(data);

cout << "Average: " << average << endl;

} catch (const exception& e) {

cerr << "Error: " << e.what() << endl;

}

} else {

cerr << "Error: Invalid data in file." << endl;

}

inputFile.close();

return 0;

}

OUTPUT

