Riphah International University

Data Structure (Lab)

Lab Task 3rd

Name: Mir Ahmad Shah

Sap id: 54906

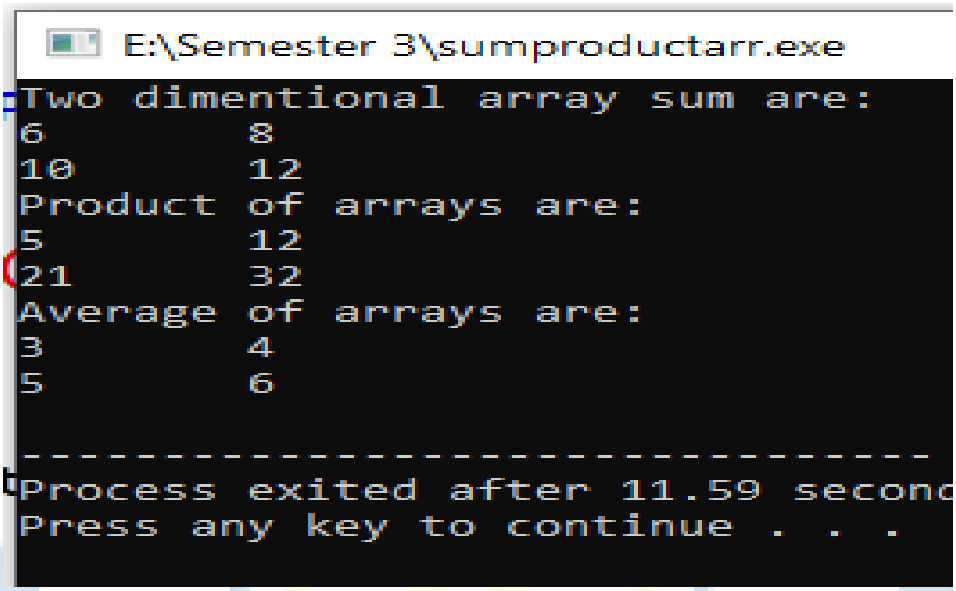
Instructor: Mr. Zeeshan Ali

Section: BSCS (3-1)

Submission Date: September 20, 2024

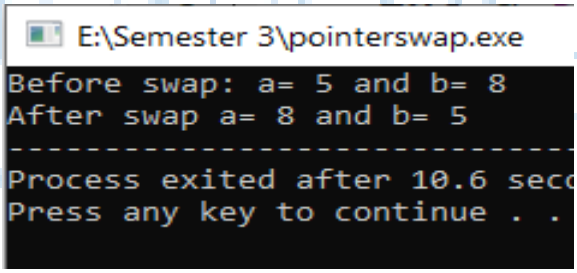
1:

```
#include <iostream>
using namespace std;

int main()
{
	int array[2][2]={{1, 2},
	                 {3, 4}};
	int array1[2][2]={{5, 6},
	                  {7, 8}};
	cout<<"Two dimentional array sum are: "<<endl;
	for(int i=0;i<2;i++)
	{
		for(int j=0;j<2;j++)
		{
			cout<<array[i][j]+array1[i][j]<<"	";
		}
		cout<<endl;

	}
	cout<<"Product of arrays are: "<<endl;
	for(int i=0;i<2;i++)
	{
		for(int j=0;j<2;j++)
		{
			cout<<array[i][j]*array1[i][j]<<"	";
		}
		cout<<endl;

	}
	cout<<"Average of arrays are: "<<endl;
	for(int i=0;i<2;i++)
	{
		for(int j=0;j<2;j++)
		{
			cout<<(array[i][j]+array1[i][j])/2<<"	";
		}
		cout<<endl;

	}
	return 0;
}
```



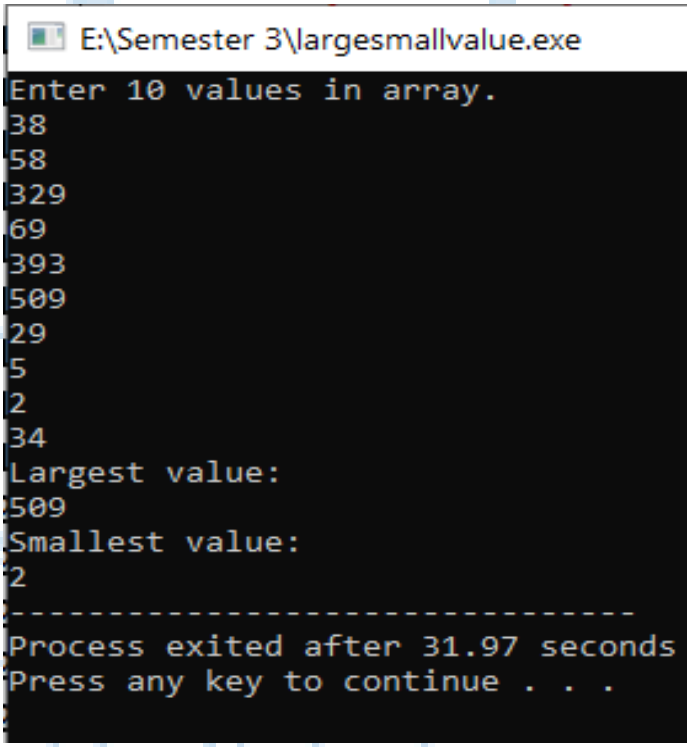
2:

```
#include <iostream>
using namespace std;

int main()
{
	int a=5;
	int b=8;
	cout<<"Before swap: a= "<<a<<" and b= "<<b<<endl;

	int *ptr1=&a;
	int *ptr2=&b;
	int *ptr3=ptr1;
	ptr1=ptr2;
	ptr2=ptr3;
	cout<<"After swap a= "<<*ptr1<<" and b= "<<*ptr2;

	return 0;
}
```



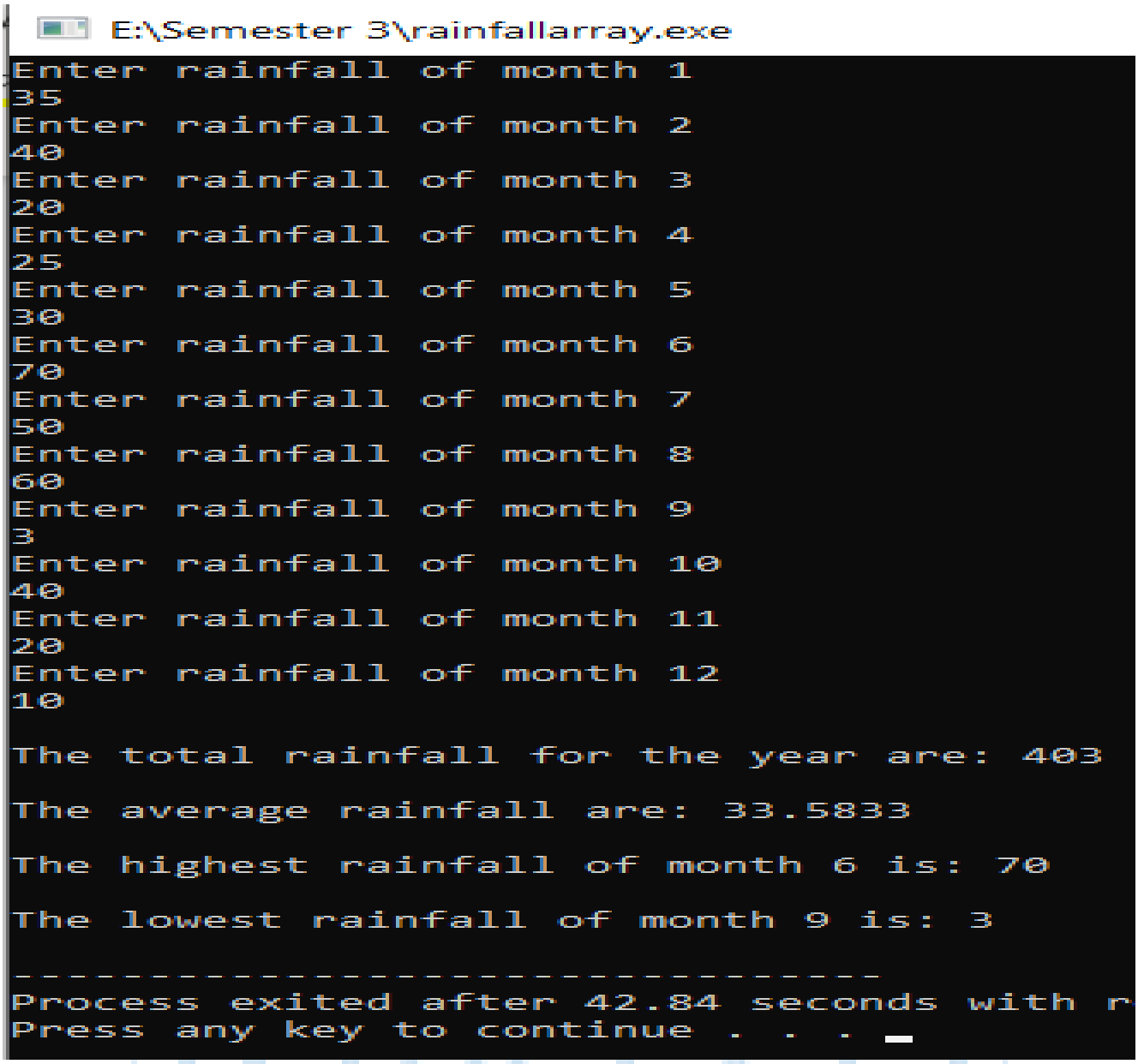
3:

```
#include <iostream>
using namespace std;

int main()
{
	int largest=0;
	int smallest;
	int arr[10];
	cout<<"Enter 10 values in array."<<endl;
	for(int i=0;i<10;i++)
	{
		cin>>arr[i];
	}
	cout<<"Largest value:"<<endl;
	for(int i=0;i<10;i++)
	{
		if(arr[i]>largest)
		{
			largest=arr[i];
		}
	}
	cout<<largest<<endl;;
	cout<<"Smallest value:"<<endl;
	for(int i=0;i<10;i++)
	{
		if(arr[i]<smallest)
		{
			smallest=arr[i];
		}
	}
	cout<<smallest;
	return 0;
}
```



4:

```
#include <iostream>
using namespace std;

int main()
{
	int a=12, lMonth, hMonth;
	double highest=0;
	double lowest=999999;
	double totalrain;
	double arr[a];
	for (int i=0;i<a;i++)
	{
		cout<<"Enter rainfall of month "<<i+1<<endl;
		cin>>arr[i];
		totalrain+=arr[i];
		if(arr[i]<lowest)
		{
			lowest=arr[i];
			lMonth=i+1;
		}
		if(arr[i]>highest)
		{
			highest=arr[i];
			hMonth=i+1;
		}
	}
	cout<<"\nThe total rainfall for the year are: "<<totalrain<<endl;
	double average=totalrain/12;
	cout<<"\nThe average rainfall are: "<<average<<endl;

	cout<<"\nThe highest rainfall of month "<<hMonth<<" is: "<<highest<<endl;
	cout<<"\nThe lowest rainfall of month "<<lMonth<<" is: "<<lowest<<endl;

	return 0;
}
```

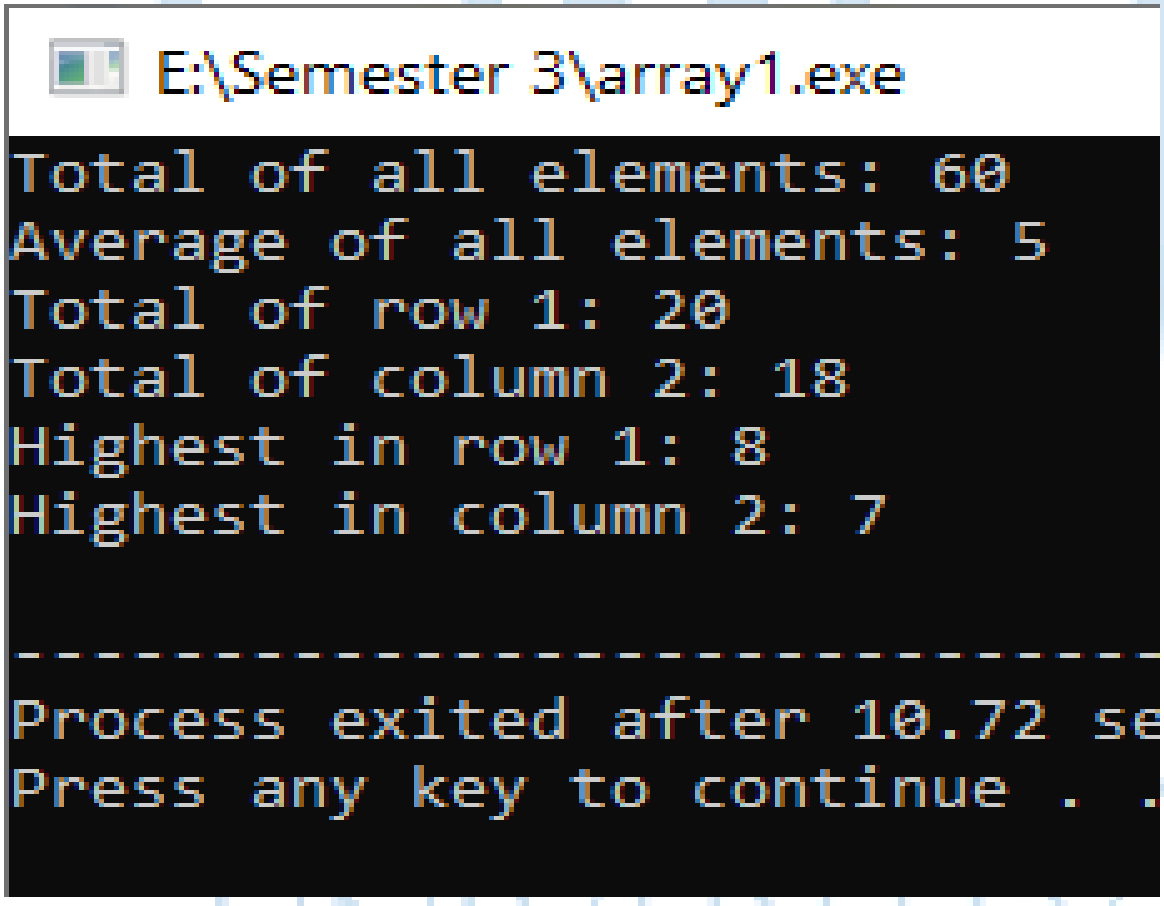


5:

```
#include <iostream>
using namespace std;

const int ROWS = 3;
const int COLS = 4;

int getTotal(int array[ROWS][COLS]) {
    int total = 0;
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            total += array[i][j];
```

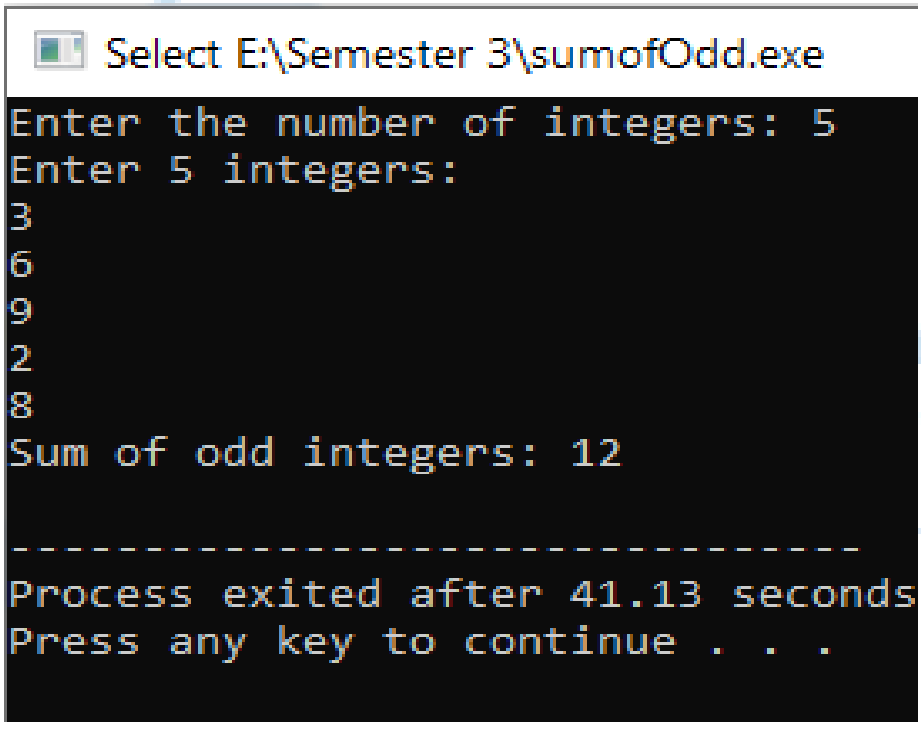
```
        }
    }
    return total;
}
double getAverage(int array[ROWS][COLS]) {
    int total = getTotal(array);
    int numElements = ROWS * COLS;
    return static_cast<double>(total) / numElements;
}
int getRowTotal(int array[ROWS][COLS], int row) {
    int rowTotal = 0;
    for (int j = 0; j < COLS; j++) {
        rowTotal += array[row][j];
    }
    return rowTotal;
}
int getColumnTotal(int array[ROWS][COLS], int col) {
    int colTotal = 0;
    for (int i = 0; i < ROWS; i++) {
        colTotal += array[i][col];
    }
    return colTotal;
}
int getHighestInRow(int array[ROWS][COLS], int row) {
    int highest = array[row][0];
    for (int j = 1; j < COLS; j++) {
        if (array[row][j] > highest) {
            highest = array[row][j];
        }
    }
    return highest;
}
int getHighestInColumn(int array[ROWS][COLS], int col) {
    int highest = array[0][col];
    for (int i = 1; i < ROWS; i++) {
        if (array[i][col] > highest) {
            highest = array[i][col];
        }
    }
    return highest;
}

int main() {
    int array[ROWS][COLS] = {
        {3, 5, 7, 9},
        {2, 4, 6, 8},
```

```
        {1, 3, 5, 7}
    };

    cout << "Total of all elements: " << getTotal(array) << endl;
    cout << "Average of all elements: " << getAverage(array) << endl;

    int row = 1;
    cout << "Total of row " << row << ": " << getRowTotal(array, row) << endl;

    int col = 2;
    cout << "Total of column " << col << ": " << getColumnTotal(array, col) << endl;

    cout << "Highest in row " << row << ": " << getHighestInRow(array, row) << endl;
    cout << "Highest in column " << col << ": " << getHighestInColumn(array, col) << endl;

    return 0;
}
```



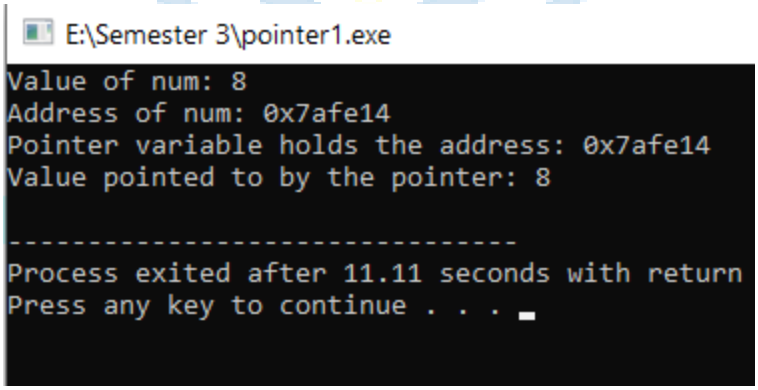
6:

```
#include <iostream>
using namespace std;

int main() {
    int size;
    cout << "Enter the number of integers: ";
    cin >> size;
    int* array = new int[size];
    cout << "Enter " << size << " integers:" << endl;
    for (int i = 0; i < size; i++) {
        cin >> array[i];
    }
    int sumOfOdd = 0;
    for (int i = 0; i < size; i++) {
        if (array[i] % 2 != 0) {
            sumOfOdd += array[i];
        }
    }
    cout << "Sum of odd integers: " << sumOfOdd << endl;
    delete[] array;

    return 0;
}
```



7:

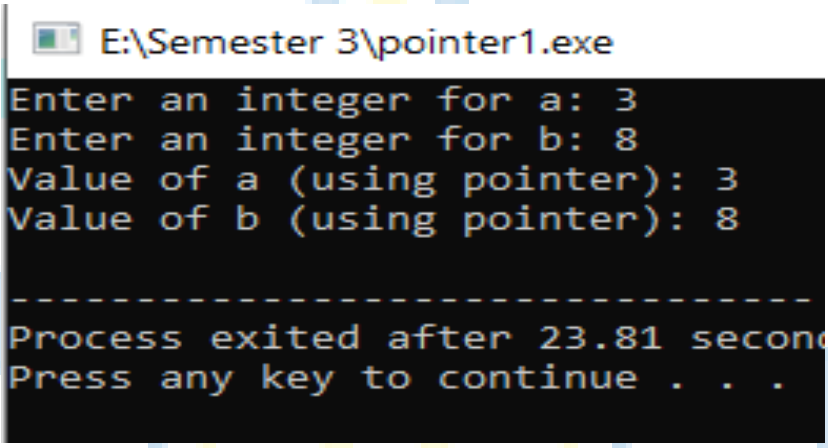
```
#include <iostream>
using namespace std;

int main() {
    int num = 8;

    int* ptr = &num;

    cout << "Value of num: " << num << endl;
    cout << "Address of num: " << &num << endl;
    cout << "Pointer variable holds the address: " << ptr << endl;
    cout << "Value pointed to by the pointer: " << *ptr << endl;

    return 0;
}
```



8:

```
#include <iostream>
using namespace std;

int main() {
    int a, b;
    cout << "Enter an integer for a: ";
    cin >> a;
    cout << "Enter an integer for b: ";
    cin >> b;
    int* ptrA = &a;
    int* ptrB = &b;
    cout << "Value of a (using pointer): " << *ptrA << endl;
    cout << "Value of b (using pointer): " << *ptrB << endl;
```

```
    return 0;
}
```



9:

```
#include <iostream>
#include <cmath>
using namespace std;

void Menu();
int Addition(int a, int b);
int Subtraction(int a, int b);
int Multiplication(int a, int b);
double Division(int a, int b);
int Pow(int number, int pow);

int main() {
    Menu();

    return 0;
}

void Menu() {
    int choice, a, b;

    do {
        cout << "Calculator Menu:" << endl;
        cout << "1. Addition" << endl;
        cout << "2. Subtraction" << endl;
        cout << "3. Multiplication" << endl;
        cout << "4. Division" << endl;
        cout << "5. Power" << endl;
```

```
        cout << "6. Exit" << endl;
        cout << "Enter your choice (1-6): ";
        cin >> choice;

        if (choice >= 1 && choice <= 4) {
            cout << "Enter two integers: ";
            cin >> a >> b;
        } else if (choice == 5) {
            cout << "Enter a base number and exponent: ";
            cin >> a >> b;
        }

        switch (choice) {
            case 1:
                cout << "Result: " << Addition(a, b) << endl;
                break;
            case 2:
                cout << "Result: " << Subtraction(a, b) << endl;
                break;
            case 3:
                cout << "Result: " << Multiplication(a, b) << endl;
                break;
            case 4:
                if (b != 0) {
                    cout << "Result: " << Division(a, b) << endl;
                } else {
                    cout << "Error: Division by zero is not allowed!" << endl;
                }
                break;
            case 5:
                cout << "Result: " << Pow(a, b) << endl;
                break;
            case 6:
                cout << "Exiting..." << endl;
                break;
            default:
                cout << "Invalid choice! Please try again." << endl;
        }

        cout << endl;
    } while (choice != 6);
}

int Addition(int a, int b) {
    return a + b;
}
```

```
int Subtraction(int a, int b) {
    return a - b;
}

int Multiplication(int a, int b) {
    return a * b;
}

double Division(int a, int b) {
    return static_cast<double>(a) / b;
}

int Pow(int number, int pow) {
    return static_cast<int>(std::pow(number, pow));
}
```