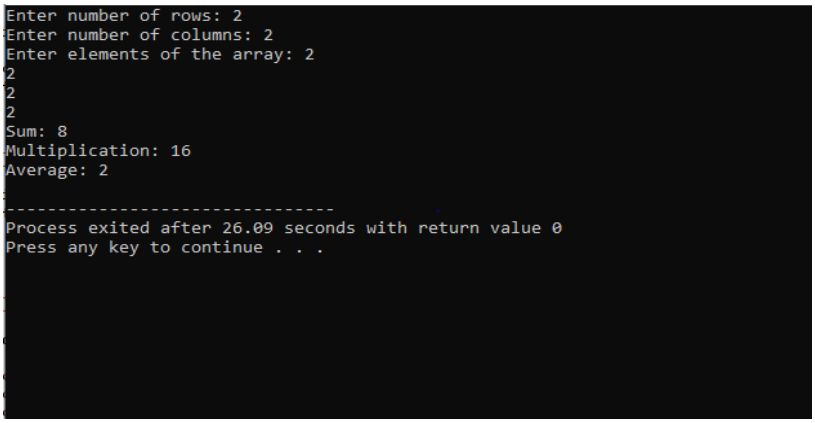
MUHAMMAD SARMAD CHUGHTAI
54915 CS3-1
DATA STRUCTURE

Lab Task 03

Question no 1:

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

int main() {
    int rows, cols;
    cout << "Enter number of rows: ";
    cin >> rows;
    cout << "Enter number of columns: ";
    cin >> cols;

    int arr[rows][cols];

    cout << "Enter elements of the array: ";
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cin >> arr[i][j];
        }
    }
    int sum = 0;
multiplication = 1;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            sum += arr[i][j];
            multiplication *= arr[i][j];
        }
    }
    double average = (double)sum / (rows * cols);

    cout << "Sum: " << sum << endl;
    cout << "Multiplication: " << multiplication << endl;
    cout << "Average: " << average << endl;

    return 0;
}
```

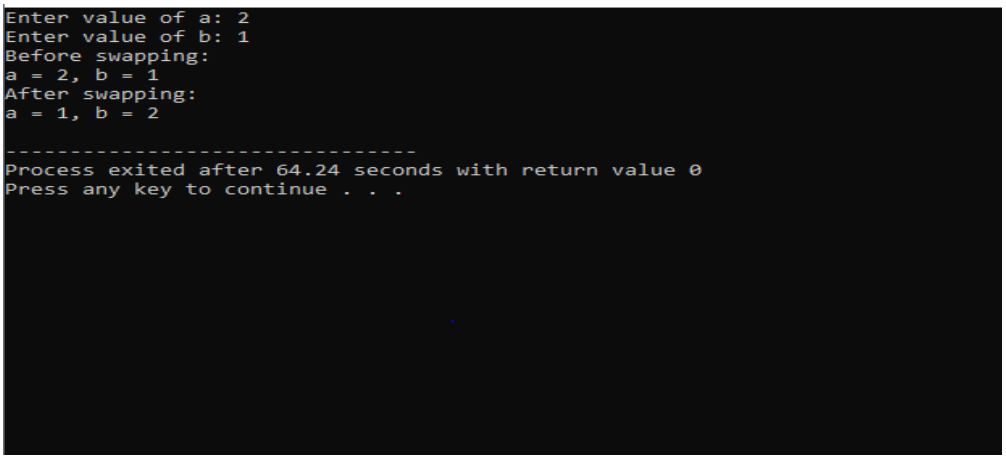
Output



Question no 2:

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

int main() {
    int a, b;
    cout << "Enter value of a: ";
    cin >> a;
    cout << "Enter value of b: ";
    cin >> b;

    int* ptrA = &a;
    int* ptrB = &b;

    cout << "Before swapping: " << endl;
    cout << "a = " << a << ", b = " << b << endl;

    // Swap values using pointers
    int temp = *ptrA;
    *ptrA = *ptrB;
    *ptrB = temp;

    cout << "After swapping: " << endl;
    cout << "a = " << a << ", b = " << b << endl;

    return 0;
}
```

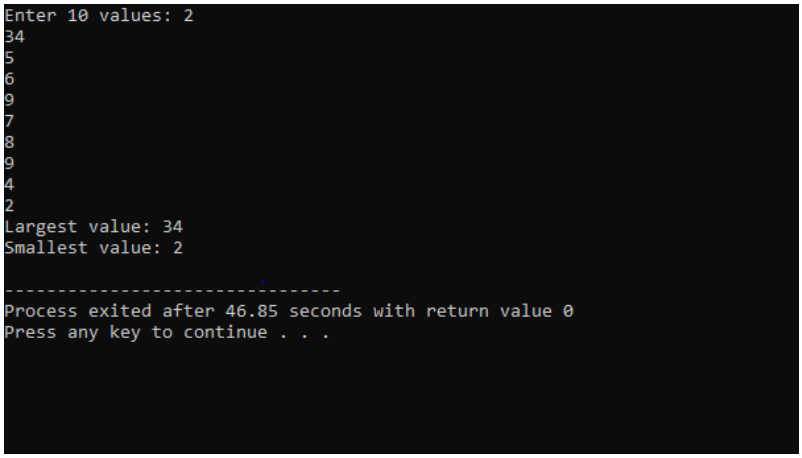
Output



Question no 3:

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

int main() {
    int arr[10];
    cout << "Enter 10 values: ";
    for (int i = 0; i < 10; i++) {
        cin >> arr[i];
    }

    int largest = arr[0], smallest = arr[0];
    for (int i = 1; i < 10; i++) {
        if (arr[i] > largest) {
            largest = arr[i];
        }
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
    }

    cout << "Largest value: " << largest << endl;
    cout << "Smallest value: " << smallest << endl;

    return 0;
}
```

Output



Question no 4:

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

int main() {
    double rainfall[12];
    cout << "Enter rainfall for each of 12 months: ";
    for (int i = 0; i < 12; i++) {
        cin >> rainfall[i];
    }

    double totalRainfall = 0;
    for (int i = 0; i < 12; i++) {
        totalRainfall += rainfall[i];
    }
```

```
    double averageRainfall = totalRainfall / 12;

    int highestMonth = 0, lowestMonth = 0;
    for (int i = 1; i < 12; i++) {
        if (rainfall[i] > rainfall[highestMonth]) {
            highestMonth = i;
        }
        if (rainfall[i] < rainfall[lowestMonth]) {
            lowestMonth = i;
        }
    }

    cout << "Total rainfall: " << totalRainfall << endl;
    cout << "Average monthly rainfall: " << averageRainfall << endl;
    cout << "Highest rainfall in month: " << highestMonth + 1 << endl;
    cout << "Lowest rainfall in month: " << lowestMonth + 1 << endl;

    return 0;
}
```

Output

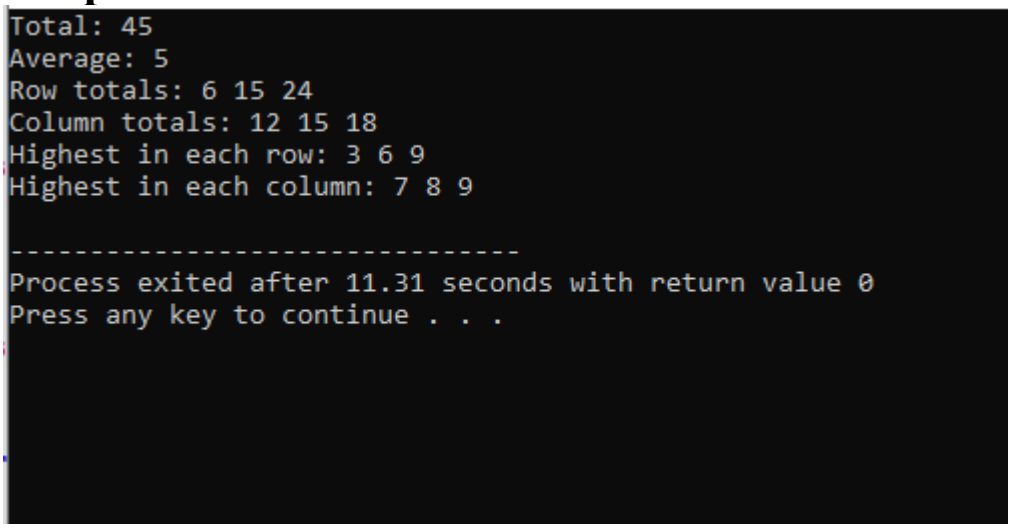
```
Enter rainfall for each of 12 months: 52
23
54
85
45
63
12
22
14
15
16
17
Total rainfall: 418
Average monthly rainfall: 34.8333
Highest rainfall in month: 4
Lowest rainfall in month: 7

--------------------------------
Process exited after 32.54 seconds with return value 0
Press any key to continue . . .
```

Question no 5:

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

int main() {
    int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

    int total = 0;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            total += arr[i][j];
        }
    }

    double average = (double)total / (3 * 3);

    cout << "Total: " << total << endl;
    cout << "Average: " << average << endl;
```

```
    int rowTotal[3] = {0, 0, 0};
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            rowTotal[i] += arr[i][j];
        }
    }

    int colTotal[3] = {0, 0, 0};
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            colTotal[j] += arr[i][j];
        }
    }

    int highestInRow[3] = {arr[0][0], arr[1][0], arr[2][0]};
    for (int i = 0; i < 3; i++) {
        for (int j = 1; j < 3; j++) {
            if (arr[i][j] > highestInRow[i]) {
                highestInRow[i] = arr[i][j];
            }
        }
    }

    int highestInCol[3] = {arr[0][0], arr[0][1], arr[0][2]};
    for (int i = 1; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (arr[i][j] > highestInCol[j]) {
                highestInCol[j] = arr[i][j];
            }
        }
    }

    cout << "Row totals: ";
    for (int i = 0; i < 3; i++) {
        cout << rowTotal[i] << " ";
    }
    cout << endl;

    cout << "Column totals: ";
    for (int i = 0; i < 3; i++) {
        cout << colTotal[i] << " ";
    }
    cout << endl;

    cout << "Highest in each row: ";
    for (int i = 0; i < 3; i++) {
        cout << highestInRow[i] << " ";
    }
    cout << endl;

    cout << "Highest in each column: ";
    for (int i = 0; i < 3; i++) {
        cout << highestInCol[i] << " ";
    }
    cout << endl;

    return 0;
}
```

Output



Question no 6:

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

int main() {
    int size;
    cout << "Enter the size of the array: ";
    cin >> size;

    int* arr = new int[size];

    cout << "Enter elements of the array: ";
    for (int i = 0; i < size; i++) {
        cin >> arr[i];
    }

    int sumOfOdd = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 != 0) {
            sumOfOdd += arr[i];
        }
    }

    cout << "Sum of odd integers: " << sumOfOdd << endl;

    return 0;
}
```

Output :

```
Enter the size of the array: 3
Enter elements of the array: 1
2
5
Sum of odd integers: 6

--------------------------------
Process exited after 25.26 seconds with return value 0
Press any key to continue . . .
```

Question no 7:

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

int main() {
    int a = 10;
    int* ptrA = &a;

    cout << "Value of a: " << a << endl;
    cout << "Address of a: " << &a << endl;
    cout << "Value of ptrA: " << ptrA << endl;
    cout << "Value at address ptrA: " << *ptrA << endl;

    return 0;
}
```

Output :

```
Value of a: 10
Address of a: 0x78fe14
Value of ptrA: 0x78fe14
Value at address ptrA: 10

--------------------------------
Process exited after 11.88 seconds with return value 0
Press any key to continue . . .
```

Question no 8:

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

int main() {
    int a, b;
    cout << "Enter value of a: ";
    cin >> a;
    cout << "Enter value of b: ";
    cin >> b;
```

```
    int* ptrA = &a;
    int* ptrB = &b;

    cout << "Value of a: " << a << endl;
    cout << "Value of b: " << b << endl;
    cout << "Value of ptrA: " << ptrA << endl;
    cout << "Value of ptrB: " << ptrB << endl;
    cout << "Value at address ptrA: " << *ptrA << endl;
    cout << "Value at address ptrB: " << *ptrB << endl;

    return 0;
}
```

Output :

```
Enter value of a: 3
Enter value of b: 4
Value of a: 3
Value of b: 4
Value of ptrA: 0x78fe0c
Value of ptrB: 0x78fe08
Value at address ptrA: 3
Value at address ptrB: 4

--------------------------------
Process exited after 23.58 seconds with return value 0
Press any key to continue . . .
```

Question no 9:

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

void menu() {
    cout << "Calculator Menu:" << endl;
    cout << "1. Addition" << endl;
    cout << "2. Subtraction" << endl;
    cout << "3. Division" << endl;
    cout << "4. Multiplication" << endl;
    cout << "5. Power" << endl;
    cout << "Enter your choice: ";
}

int addition(int a, int b) {
    return a + b;
}

int subtraction(int a, int b) {
    return a - b;
}

double division(int a, int b) {
    if (b == 0) {
        cout << "Error: Division by zero!" << endl;
        return 0;
    }
```

```
    return (double)a / b;
}

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

double power(int number, int pow) {
    return power(number, pow);
}

int main() {
    menu();
    int choice;
    cin >> choice;

    int a, b;
    cout << "Enter two numbers: ";
    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: " << division(a, b) << endl;
            break;
        case 4:
            cout << "Result: " << multiplication(a, b) << endl;
            break;
        case 5:
            cout << "Result: " << power(a, b) << endl;
            break;
        default:
            cout << "Invalid choice!" << endl;
    }

    return 0;
}
```

OUTPUT