LAB TASK 2:

Name: Saif Majid Khan

SAP-ID: 57114

CS3-1

Data Structures.

Q1)

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

int main () {

int rows = 2;
int cols = 2;
int array[rows][cols];
cout << "Enter the elements of the " << rows << "X" << cols
<<endl;
```

```cpp
int sum = 0;
int product = 1;
for (int i=0; i < rows; ++1) {    // Nest

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

cout << "Element [" << i << "][" << j << "]:  ";
cin >> array[i][j];
sum += array[i][j];
product *= array[i][j];
}
 }
```

```cpp
double average = static_cast(sum) / (rows * cols);

cout << "Sum of all elements: " << sum << endl;
cout << "Product of all elements: " << product << endl;
cout << "Average of all elements: " << average << endl;

return 0;

 }
```

```
Enter the elements of the 2X2
Element [0][0]: 1
Element [0][1]: 2
Element [0][2]: 3
Element [0][3]: 4
Element [0][4]: 5
Element [0][5]: 6
Element [0][6]: 7
Sum of all elements: 16
Product of all elements: 0
Average of all elements: 4


...Program finished with exit code 0
Press ENTER to exit console.
```

Q2)

```cpp
#include <iostream>
using namespace std;
 int main() {
int x, y;,

// Input values for x and y

cout << "Enter value of x: ";
cin >> x;

cout << "Enter value of y: ";
cin >> y;
```

```cpp
cout << "Before swapping: x = " << x << " y = " << y << endl;
 // Swap the values

int temp = x;
x = y;
y = temp;

 cout << "After swapping: x = " << x << " y = " << y << endl;

return 0;
}
```

```
Enter value of x: 3
Enter value of y: 2
Before swapping: x = 3 y = 2
After swapping: x = 2 y = 3


...Program finished with exit code 0
Press ENTER to exit console.
```

Q3)

```cpp
#include <iostream>
using namespace std;
int main() {
int size = 10;
 int values[size];
cout << "Enter 10 integer values: " << endl;

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

cin >> values[i]; }
```

```
int largest = values[0]; // Dec..
int smallest = values[0];

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

if (values[i] > largest) {

largest = values[i];
 }
if (values[i] < smallest) {

smallest = values[i];
    }
 }
```

```cpp
cout << "Largest value: " << largest << endl;
cout << "Smallest value: " << smallest << endl;
return 0;
}
```

```
Enter 10 integer values:
1
2
3
4
5
6
7
8
9
10
Largest value: 10
Smallest value: 1


...Program finished with exit code 0
Press ENTER to exit console.
```

Q4)

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

int main() {

    int MONTHS = 12;
    double rainfall[MONTHS];
    double totalRainfall = 0.0;
    double averageRainfall;
    int monthWithHighest = 0;
    int monthWithLowest = 0;
```

```cpp
 cout << "Enter the total rainfall for each of the 12 months (in inches): "
<< endl;
    for (int i = 0; i < MONTHS; ++i) {
        cout << "Month " << (i + 1) << ": ";
        cin >> rainfall[i];
        totalRainfall += rainfall[i];

        if (rainfall[i] > rainfall[monthWithHighest]) {
            monthWithHighest = i;
        }
        if (i == 0 || rainfall[i] < rainfall[monthWithLowest]) { // Fixed
initialization
            monthWithLowest = i;
        }
    }
```

```cpp
averageRainfall = totalRainfall / MONTHS;

    cout << "Total rainfall for the year: " << totalRainfall << " inches" <<
endl;
    cout << "Average monthly rainfall: " << averageRainfall << " inches"
<< endl;
    cout << "Month with highest rainfall: Month " << (monthWithHighest +
1) << " with " << rainfall[monthWithHighest] << " inches" << endl;
    cout << "Month with lowest rainfall: Month " << (monthWithLowest +
1) << " with " << rainfall[monthWithLowest] << " inches" << endl;

    return 0;
}
```

```
Enter the total rainfall for each of the 12 months (in inches):
Month 1: 12
Month 2: 22
Month 3: 11
Month 4: 14
Month 5: 16
Month 6: 15
Month 7: 22
Month 8: 44
Month 9: 21
Month 10: 42
Month 11: 42
Month 12: 56
Total rainfall for the year: 317 inches
Average monthly rainfall: 26.4167 inches
Month with highest rainfall: Month 12 with 56 inches
Month with lowest rainfall: Month 3 with 11 inches


...Program finished with exit code 0
Press ENTER to exit console.
```

Q5)

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

const int ROWS = 3, COLS = 4;

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

```
int getRowTotal(int arr[ROWS][COLS], int row) {
    int total = 0;
    for (int j = 0; j < COLS; ++j)
        total += arr[row][j];
    return total;
}

int getColumnTotal(int arr[ROWS][COLS], int col) {
    int total = 0;
    for (int i = 0; i < ROWS; ++i)
        total += arr[i][col];
    return total;
}
```

```
int getHighestInRow(int arr[ROWS][COLS], int row) {
    int highest = arr[row][0];
    for (int j = 1; j < COLS; ++j)
        highest = max(highest, arr[row][j]);
    return highest;
}
int getHighestInColumn(int arr[ROWS][COLS], int col) {
    int highest = arr[0][col];
    for (int i = 1; i < ROWS; ++i)
        highest = max(highest, arr[i][col]);
    return highest;
}
int main() {
    int data[ROWS][COLS] = {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
        {9, 10, 11, 12}  };
```

```cpp
int row, col;

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

    cout << "Enter row (0-" << ROWS-1 << ") to get row total: ";
    cin >> row;
    if (row >= 0 && row < ROWS)
        cout << "Total of row " << row << ": " << getRowTotal(data, row) << endl;

    cout << "Enter column (0-" << COLS-1 << ") to get column total: ";
    cin >> col;
    if (col >= 0 && col < COLS)
        cout << "Total of column " << col << ": " << getColumnTotal(data, col) <<
endl;
```

```cpp
 cout << "Enter row (0-" << ROWS-1 << ") to get highest in row: ";
    cin >> row;
    if (row >= 0 && row < ROWS)
        cout << "Highest in row " << row << ": " << getHighestInRow(data, row)
<< endl;

    cout << "Enter column (0-" << COLS-1 << ") to get highest in column: ";
    cin >> col;
    if (col >= 0 && col < COLS)
        cout << "Highest in column " << col << ": " <<
getHighestInColumn(data, col) << endl;

    return 0;
}
```

```
Total of all elements: 78
Average of all elements: 6.5
Enter row (0-2) to get row total: 1
Total of row 1: 26
Enter column (0-3) to get column total: 2
Total of column 2: 21
Enter row (0-2) to get highest in row: 2
Highest in row 2: 12
Enter column (0-3) to get highest in column: 1
Highest in column 1: 10


...Program finished with exit code 0
Press ENTER to exit console.
```

Q6)

```cpp
#include <iostream>
using namespace std;
int main() {
 int n;
 cout << "Enter the number of elements: ";
 cin >> n;
 int* arr = new int[n];
 cout << "Enter " << n << " integers: "<< endl;
 for (int i = 0; i < n; ++i) {
 cin >> arr[i];
 }
```

```cpp
int sumOdd = 0;
 for (int i = 0; i < n; ++i) {
 if (arr[i] % 2 != 0) {
 sumOdd += arr[i];
 }
 }
 cout << "Sum of odd integers: " << sumOdd << endl;
 delete[] arr;
 return 0;
}
```

```
Enter the number of elements: 3
Enter 3 integers:
1
2
3
Sum of odd integers: 4


...Program finished with exit code 0
Press ENTER to exit console.
```
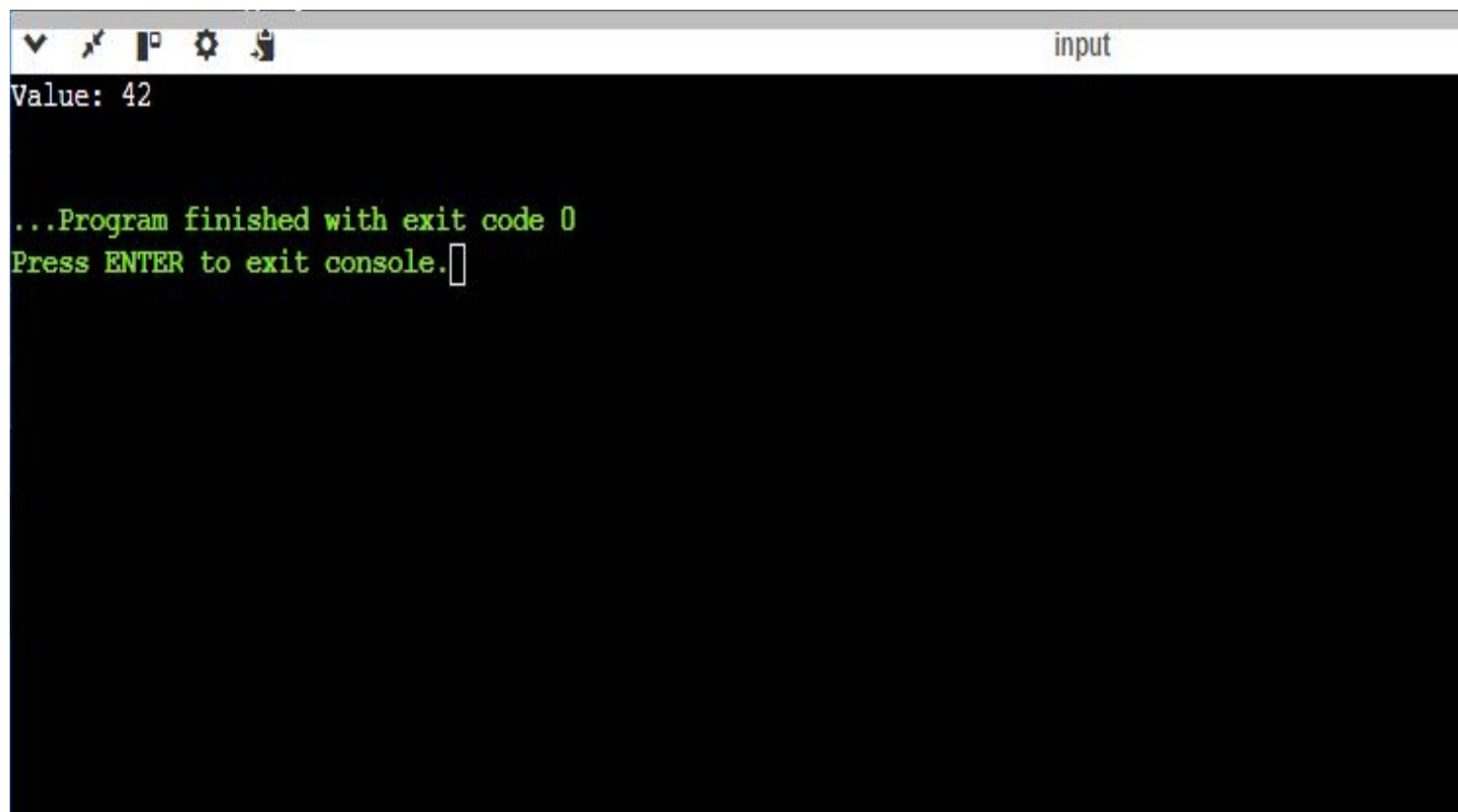
Q7)

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

int main() {
    int value = 42; // Define an integer variable
    int* ptr = &value; // Define a pointer and assign it the address of 'value'

    cout << "Value: " << *ptr << endl; // Dereference pointer to access the value

 return 0;
}
```

```
Value: 42



...Program finished with exit code 0
Press ENTER to exit console.
```

Q8)

```cpp
#include <iostream>
using namespace std;
int main() {
 int a, b; // Variables to store user inputs
 int* ptrA = &a; // Pointer to store the address of 'a'
 int* ptrB = &b; // Pointer to store the address of 'b'
 cout << "Enter integer for a: ";
 cin >> a;
 cout << "Enter integer for b: ";
 cin >> b;
 cout << "Value of a through pointer: " << *ptrA << endl;
 cout << "Value of b through pointer: " << *ptrB << endl;
 return 0; }
```

```
Enter integer for a: 2
Enter integer for b: 1
Value of a through pointer: 2
Value of b through pointer: 1


...Program finished with exit code 0
Press ENTER to exit console.
```

Q9)

```cpp
#include <iostream>
using namespace std;
// Function to display the menu and perform operations
void Menu() {
 int choice, a, b;

do {
 cout << "Calculator Menu:\n";
 cout << "1. Addition\n";
 cout << "2. Subtraction\n";
 cout << "3. Division\n";
 cout << "4. Multiplication\n";
 cout << "5. Power\n";
 cout << "6. Exit\n";
 cout << "Enter your choice: ";
 cin >> choice;
```

```cpp
if (choice >= 1 && choice <= 5) {
 cout << "Enter two integers: ";
 cin >> a >> b;
 }
 switch (choice) {
 case 1:
 cout << "Result: " << (a + b) << endl;
 break;
 case 2:
 cout << "Result: " << (a - b) << endl;
 break;
case 3:
 if (b != 0)
 cout << "Result: " << (static_cast<double>(a) / b) << endl;
 else
 cout << "Error: Division by zero" << endl;
 break;
```

```
case 4:
 cout << "Result: " << (a * b) << endl;
 break;
 case 5: {
 int pow = 1;
 for (int i = 0; i < b; ++i)
 pow *= a;
 cout << "Result: " << pow << endl;
 break;
 }
```

```cpp
case 6:
 cout << "Exiting...\n";
 break;
 default:
 cout << "Invalid choice\n";
 }
 } while (choice != 6);
}
int main() {
 Menu(); // Call the menu function
 return 0;
}
```

```
Calculator Menu:
1. Addition
2. Subtraction
3. Division
4. Multiplication
5. Power
6. Exit
Enter your choice: 4
Enter two integers: 5
5
Result: 25
Calculator Menu:
1. Addition
2. Subtraction
3. Division
4. Multiplication
5. Power
6. Exit
Enter your choice: 6
Exiting...


...Program finished with exit code 0
Press ENTER to exit console.
```