Triton SS & College

Seti Opi Marga Koteshwor-32, Kathmandu

Lab Report Of Computer Science on C Programming Lab Report No: 05

Submitted By:

Name: Rama Shankar Rai
Roll:32

Class11

Sec:503(PHY)

Submitted to:

Computer Science Department

Triton SS and College

Lab-05: [1D and 2D Array]

Introduction to 1D and 2D Arrays in C

In the C programming language, arrays are used to store multiple values of the same data type in a single variable. They are particularly useful when dealing with large datasets and can be thought of as collections of variables. Arrays in C can be one-dimensional (1D) or two-dimensional (2D).

Objectives:

1. **Understand the Concept of Arrays**: Learn what arrays are and how they work in C.
2. **Declare and Initialize Arrays**: Learn how to declare and initialize both 1D and 2D arrays.
3. **Access and Manipulate Array Elements**: Understand how to access and modify array elements.
4. **Implement Array Operations**: Implement various operations such as traversal, insertion, and deletion on arrays.

Theory:

1D Arrays:

- **Definition**: A one-dimensional array is a list of variables of the same type stored in contiguous memory locations.
- **Declaration**: A 1D array is declared as follows:

```
dataType arrayName[arraySize];
```

For example:

```
int numbers[10];
```

This declares an array named `numbers` that can hold 10 integers.

- **Initialization**: You can initialize an array when you declare it:

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

- **Accessing Elements**: Array elements are accessed using the array index. The index of the first element is 0:

```
int firstNumber = numbers[0];
```

2D Arrays:

- **Definition**: A two-dimensional array is an array of arrays. It can be visualized as a matrix with rows and columns.
- **Declaration**: A 2D array is declared as follows

```
dataType arrayName[rows][columns];
```

For example:

```
int matrix[3][4];
```

This declares a 2D array named `matrix` with 3 rows and 4 columns.

- **Initialization**: You can initialize a 2D array when you declare it:

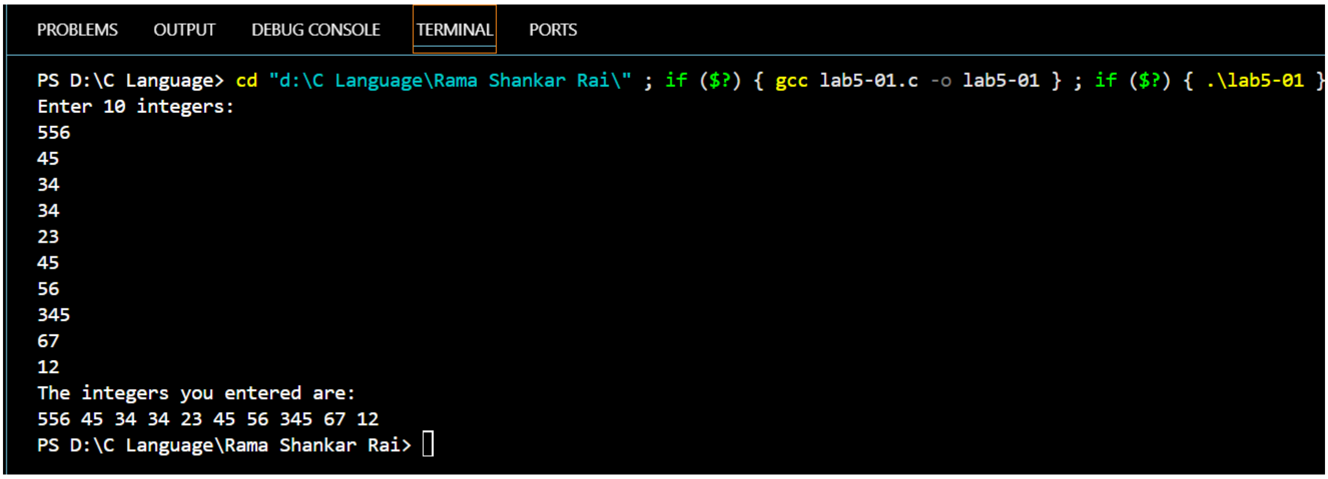
```
int matrix[2][3] = {
    {1, 2, 3},
    {4, 5, 6}
};
```

- **Accessing Elements**: Elements in a 2D array are accessed using two indices, one for the row and one for the column:

```
int element = matrix[1][2]; // Accesses the element at second row,
third
```

1.Write a program to input 10 integers into an array and display them.

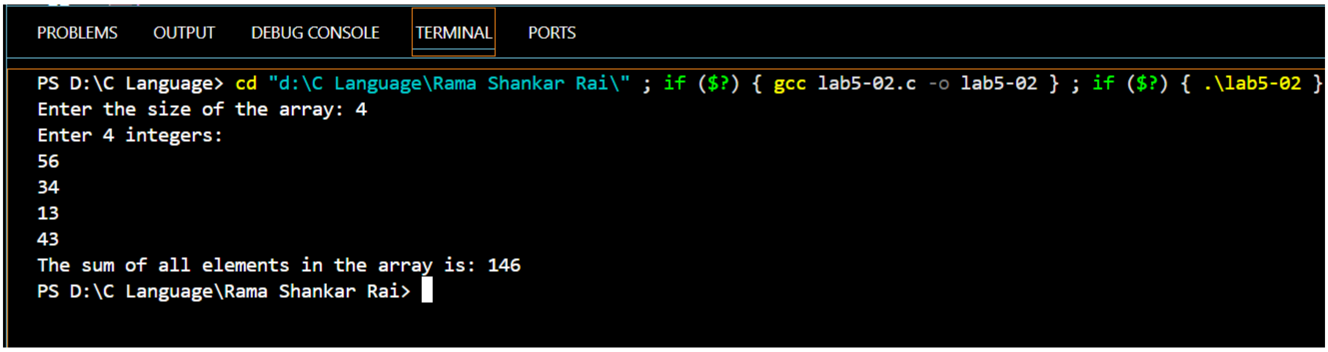
```
#include <stdio.h>
int main()
{
   int arr[10];
   int i;
   // Input 10 integers
   printf("Enter 10 integers:\n");
   for(i = 0; i < 10; i++) {
   while (scanf("%d", &arr[i]) != 1) {
   printf("Invalid input. Please enter an integer:\n");
   while (getchar() != '\n'); // clear the invalid input
}
}
// Display the integers
printf("The integers you entered are:\n");
for(i = 0; i < 10; i++) {
printf("%d ", arr[i]);
}
return 0;

}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-01.c -o lab5-01 } ; if ($?) { .\lab5-01 }
Enter 10 integers:
556
45
34
34
23
45
56
345
67
12
The integers you entered are:
556 45 34 34 23 45 56 345 67 12
PS D:\C Language\Rama Shankar Rai> 
```

2.Write a program to find the sum of all elements in an array of size n.

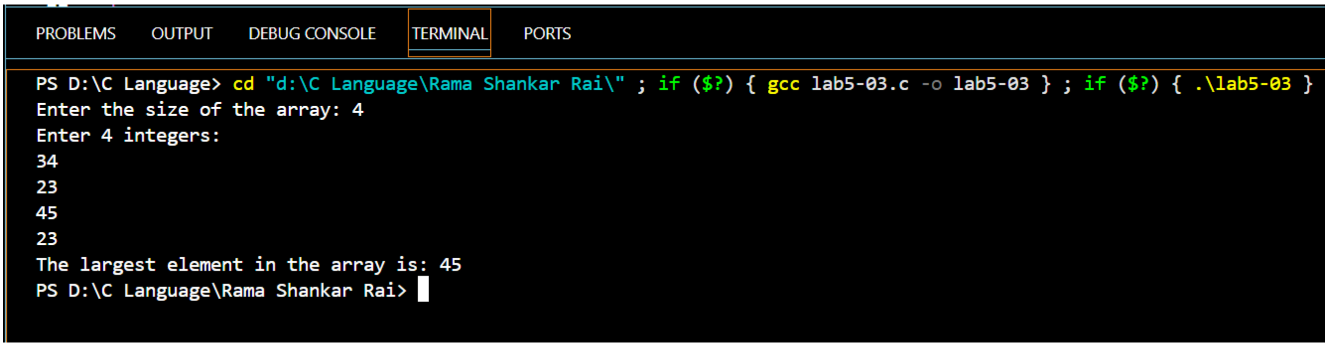
```
#include<stdio.h>
int main()
{
int n, i, sum = 0;
printf("Enter the size of the array: ");
scanf("%d", &n);
int arr[n];
printf("Enter %d integers:\n", n);
for(i = 0; i < n; i++) {
scanf("%d", &arr[i]);
}
for(i = 0; i < n; i++) {
sum += arr[i];
}
printf("The sum of all elements in the array is: %d\n",
sum);
return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-02.c -o lab5-02 } ; if ($?) { .\lab5-02 }
Enter the size of the array: 4
Enter 4 integers:
56
34
13
43
The sum of all elements in the array is: 146
PS D:\C Language\Rama Shankar Rai>
```

3.Write a program to find the largest element in an array of size n.

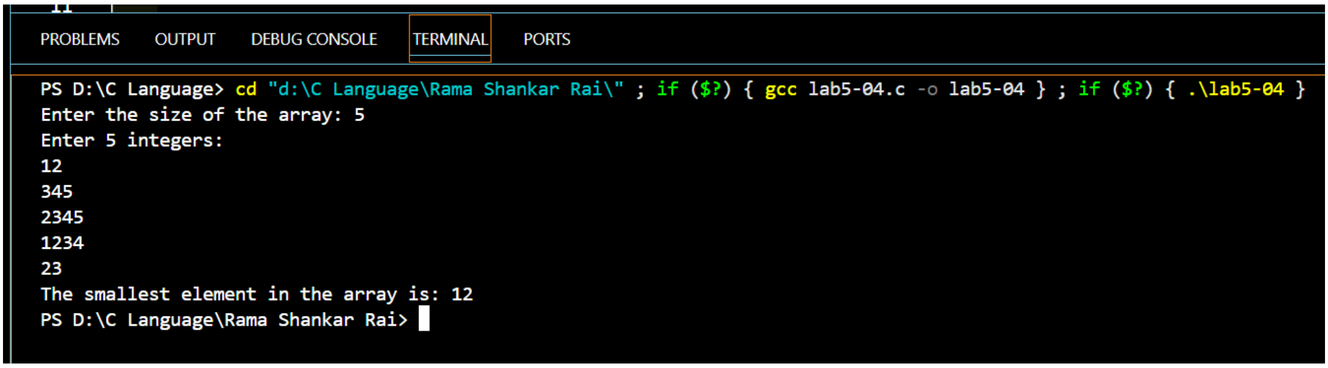
```
#include<stdio.h>

int main() {
    int n, i, max;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    max = arr[0]; // Assume the first element is the largest initially

    // Find the largest element in the array
    for(i = 1; i < n; i++) {
        if(arr[i] > max) {
            max = arr[i];
        }
    }

    // Print the largest element in the array
    printf("The largest element in the array is: %d\n", max);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-03.c -o lab5-03 } ; if ($?) { .\lab5-03 }
Enter the size of the array: 4
Enter 4 integers:
34
23
45
23
The largest element in the array is: 45
PS D:\C Language\Rama Shankar Rai> 
```

4.Write a program to find the smallest element in an array of size n.

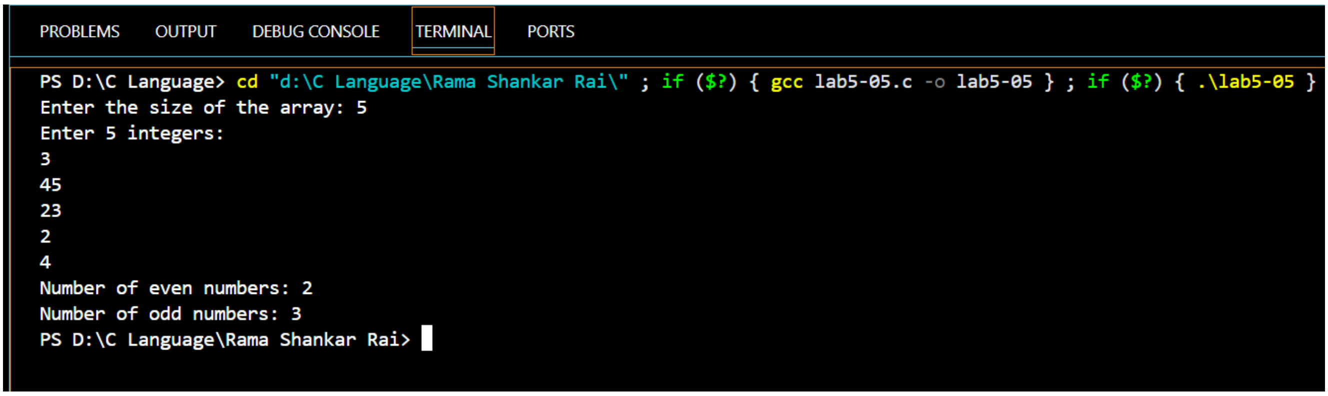
```
#include<stdio.h>

int main() {
    int n, i, min;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    min = arr[0]; // Assume the first element is the smallest initially

    // Find the smallest element in the array
    for(i = 1; i < n; i++) {
        if(arr[i] < min) {
            min = arr[i];
        }
    }

    // Print the smallest element in the array
    printf("The smallest element in the array is: %d\n", min);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-04.c -o lab5-04 } ; if ($?) { .\lab5-04 }
Enter the size of the array: 5
Enter 5 integers:
12
345
2345
1234
23
The smallest element in the array is: 12
PS D:\C Language\Rama Shankar Rai> 
```

5.Write a program to count the number of even and odd numbers in an array.

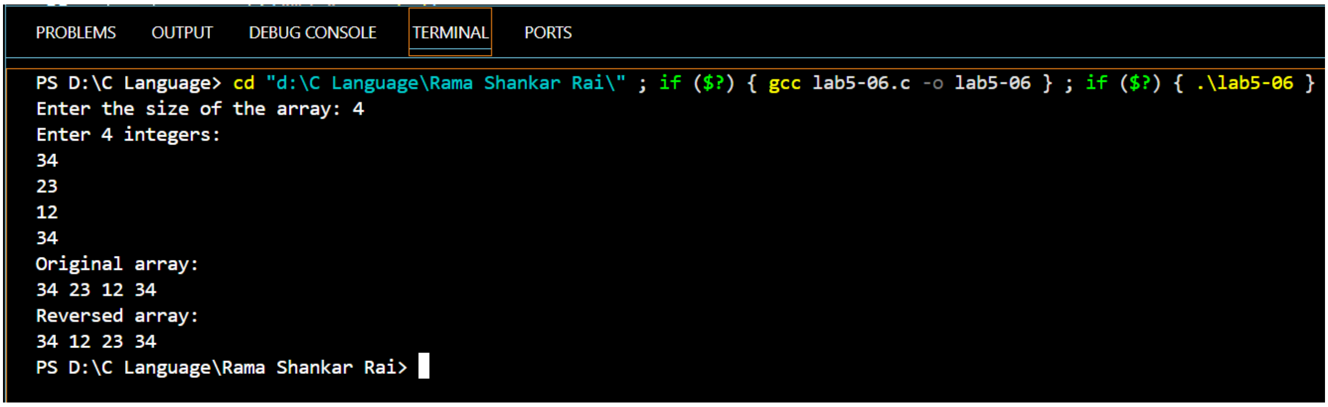
```
#include<stdio.h>

int main() {
   int n, i, evenCount = 0, oddCount = 0;

   // Input the size of the array
   printf("Enter the size of the array: ");
   scanf("%d", &n);

   int arr[n];

   // Input the elements of the array
   printf("Enter %d integers:\n", n);
   for(i = 0; i < n; i++) {
      scanf("%d", &arr[i]);
   }

   // Count even and odd numbers
   for(i = 0; i < n; i++) {
      if(arr[i] % 2 == 0) {
         evenCount++;
      } else {
         oddCount++;
      }
   }

   // Print the counts of even and odd numbers
   printf("Number of even numbers: %d\n", evenCount);
   printf("Number of odd numbers: %d\n", oddCount);

   return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-05.c -o lab5-05 } ; if ($?) { .\lab5-05 }
Enter the size of the array: 5
Enter 5 integers:
3
45
23
2
4
Number of even numbers: 2
Number of odd numbers: 3
PS D:\C Language\Rama Shankar Rai>
```

6.Write a program to reverse an array of size n and display the reversed array.

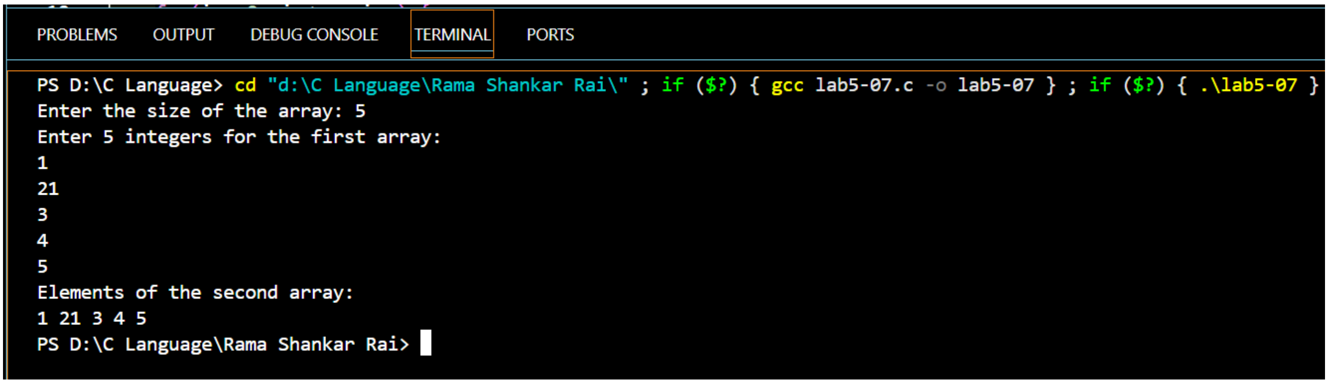
```c
#include<stdio.h>

int main() {
    int n, i;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Original array:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    // Reverse the array
    printf("Reversed array:\n");
    for(i = n-1; i >= 0; i--) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-06.c -o lab5-06 } ; if ($?) { .\lab5-06 }
Enter the size of the array: 4
Enter 4 integers:
34
23
12
34
Original array:
34 23 12 34
Reversed array:
34 12 23 34
PS D:\C Language\Rama Shankar Rai>
```

7.Write a program to copy the elements of one array into another array.

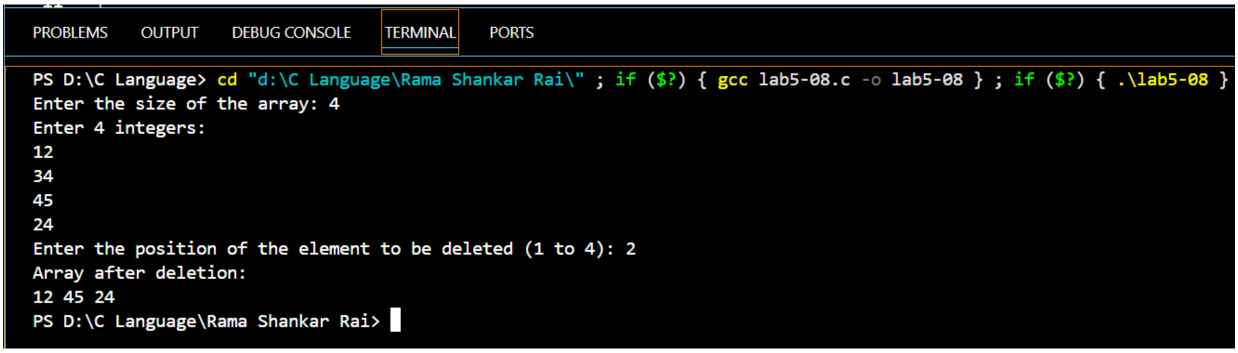
```
#include<stdio.h>

int main() {
    int n, i;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr1[n], arr2[n];

    // Input the elements of the first array
    printf("Enter %d integers for the first array:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr1[i]);
    }

    // Copy elements from arr1 to arr2
    for(i = 0; i < n; i++) {
        arr2[i] = arr1[i];
    }

    // Print the elements of the second array
    printf("Elements of the second array:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr2[i]);
    }
    printf("\n");

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-07.c -o lab5-07 } ; if ($?) { .\lab5-07 }
Enter the size of the array: 5
Enter 5 integers for the first array:
1
21
3
4
5
Elements of the second array:
1 21 3 4 5
PS D:\C Language\Rama Shankar Rai>
```

8.Write a program to delete an element from a specific position in an array.

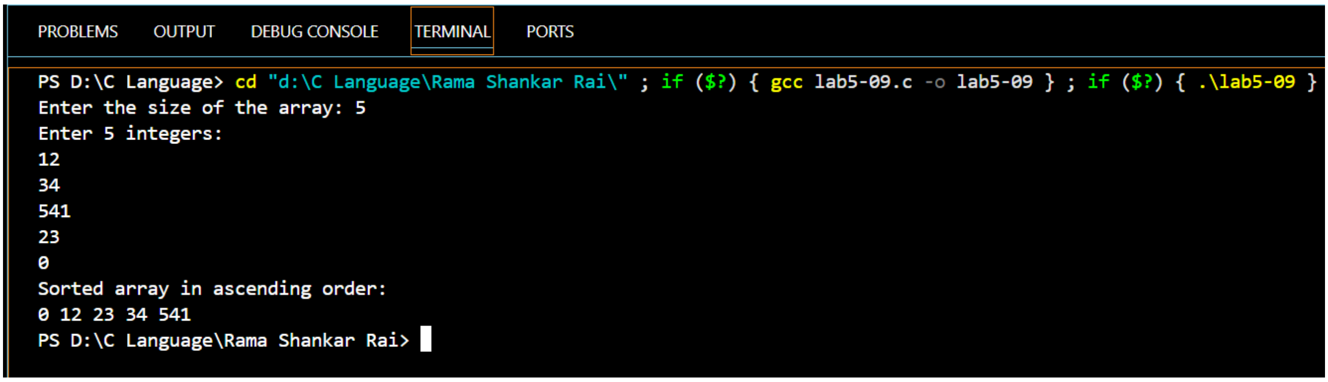
```
#include<stdio.h>

int main() {
    int n, i, pos;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Input the position from which the element should be deleted
    printf("Enter the position of the element to be deleted (1 to %d): ", n);
    scanf("%d", &pos);

    // Check if the position is valid
    if(pos < 1 || pos > n) {
        printf("Invalid position!\n");
        return 1;
    }

    // Shift elements to the left to delete the element
    for(i = pos - 1; i < n - 1; i++) {
        arr[i] = arr[i + 1];
    }

    // Decrement the size of the array
    n--;

    // Print the array after deletion
    printf("Array after deletion:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-08.c -o lab5-08 } ; if ($?) { .\lab5-08 }
Enter the size of the array: 4
Enter 4 integers:
12
34
45
24
Enter the position of the element to be deleted (1 to 4): 2
Array after deletion:
12 45 24
PS D:\C Language\Rama Shankar Rai> 
```

9.Write a program to sort an array of integers in ascending order.

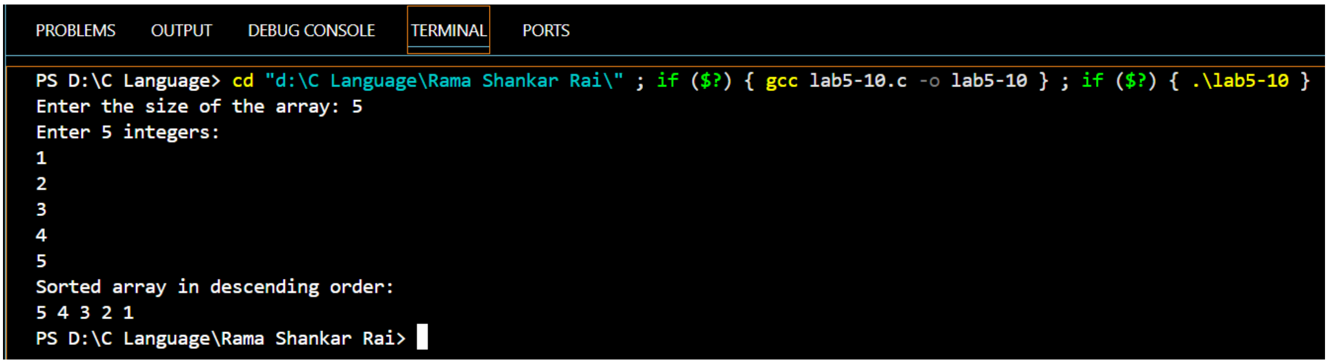
```
#include<stdio.h>

int main() {
    int n, i, j, temp;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Bubble Sort algorithm to sort the array in ascending order
    for(i = 0; i < n - 1; i++) {
        for(j = 0; j < n - i - 1; j++) {
            if(arr[j] > arr[j + 1]) {
                // Swap arr[j] and arr[j + 1]
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    // Print the sorted array
    printf("Sorted array in ascending order:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-09.c -o lab5-09 } ; if ($?) { .\lab5-09 }
Enter the size of the array: 5
Enter 5 integers:
12
34
541
23
0
Sorted array in ascending order:
0 12 23 34 541
PS D:\C Language\Rama Shankar Rai> 
```

10.Write a program to sort an array of integers in descending order.

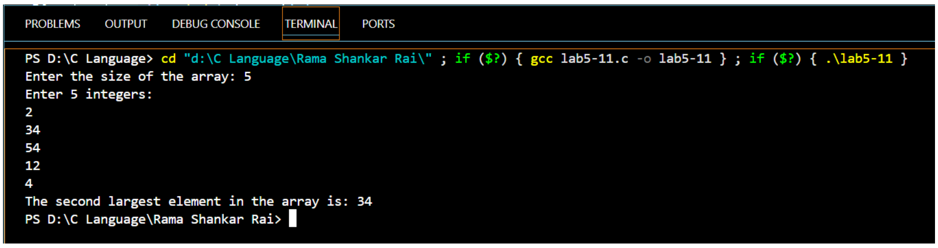
```
#include<stdio.h>

int main() {
    int n, i, j, temp;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Bubble Sort algorithm to sort the array in descending order
    for(i = 0; i < n - 1; i++) {
        for(j = 0; j < n - i - 1; j++) {
            if(arr[j] < arr[j + 1]) {
                // Swap arr[j] and arr[j + 1]
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    // Print the sorted array
    printf("Sorted array in descending order:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-10.c -o lab5-10 } ; if ($?) { .\lab5-10 }
Enter the size of the array: 5
Enter 5 integers:
1
2
3
4
5
Sorted array in descending order:
5 4 3 2 1
PS D:\C Language\Rama Shankar Rai> 
```

11. Write a program to find the second largest element in an array.

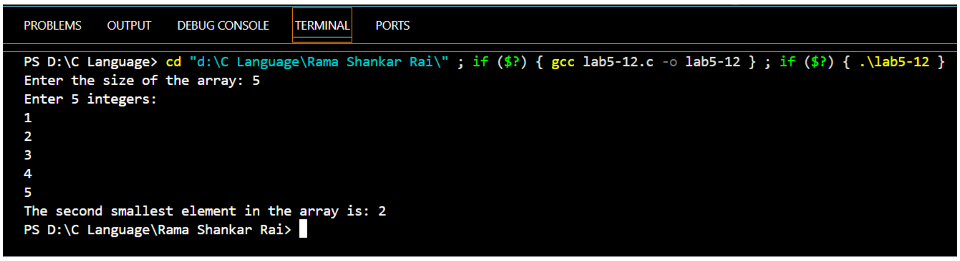
```
#include<stdio.h>

int main() {
    int n, i, largest, secondLargest;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    if(n < 2) {
        printf("Array must have at least two elements to find the second largest element.\n");
        return 1;
    }

    // Initialize the largest and second largest elements
    if(arr[0] > arr[1]) {
        largest = arr[0];
        secondLargest = arr[1];
    } else {
        largest = arr[1];
        secondLargest = arr[0];
    }

    // Find the largest and second largest elements in the array
    for(i = 2; i < n; i++) {
        if(arr[i] > largest) {
            secondLargest = largest;
            largest = arr[i];
        } else if(arr[i] > secondLargest && arr[i] != largest) {
            secondLargest = arr[i];
        }
    }

    // Print the second largest element in the array
    printf("The second largest element in the array is: %d\n", secondLargest);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-11.c -o lab5-11 } ; if ($?) { .\lab5-11 }
Enter the size of the array: 5
Enter 5 integers:
2
34
54
12
4
The second largest element in the array is: 34
PS D:\C Language\Rama Shankar Rai>
```

12. Write a program to find the second smallest element in an array.

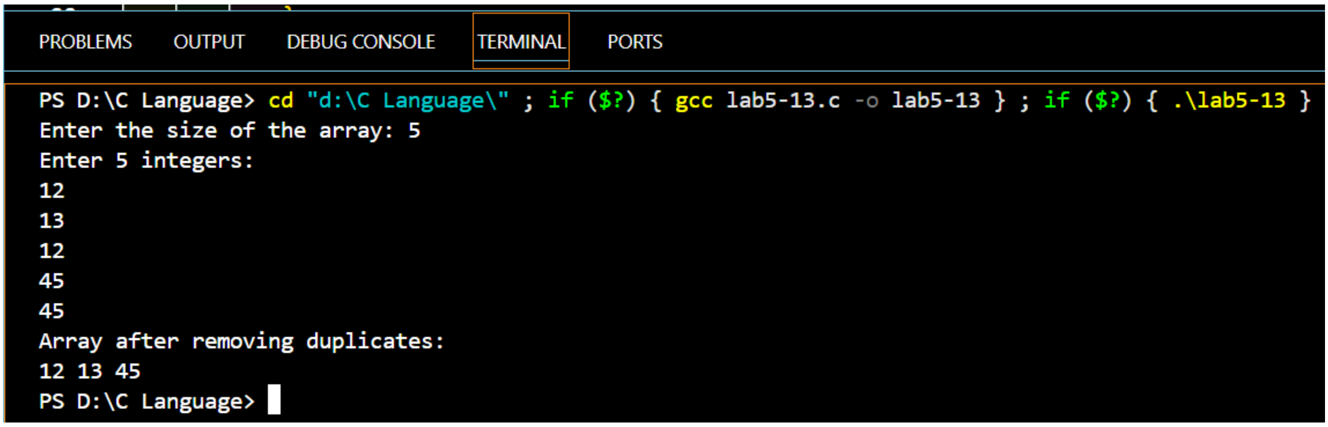
```c
#include<stdio.h>

int main() {
    int n, i, smallest, secondSmallest;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    if(n < 2) {
        printf("Array must have at least two elements to find the second smallest element.\n");
        return 1;
    }

    // Initialize the smallest and second smallest elements
    if(arr[0] < arr[1]) {
        smallest = arr[0];
        secondSmallest = arr[1];
    } else {
        smallest = arr[1];
        secondSmallest = arr[0];
    }

    // Find the smallest and second smallest elements in the array
    for(i = 2; i < n; i++) {
        if(arr[i] < smallest) {
            secondSmallest = smallest;
            smallest = arr[i];
        } else if(arr[i] < secondSmallest && arr[i] != smallest) {
            secondSmallest = arr[i];
        }
    }

    // Print the second smallest element in the array
    printf("The second smallest element in the array is: %d\n", secondSmallest);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\Rama Shankar Rai\" ; if ($?) { gcc lab5-12.c -o lab5-12 } ; if ($?) { .\lab5-12 }
Enter the size of the array: 5
Enter 5 integers:
1
2
3
4
5
The second smallest element in the array is: 2
PS D:\C Language\Rama Shankar Rai>
```

13. Write a program to remove duplicate elements from an array.

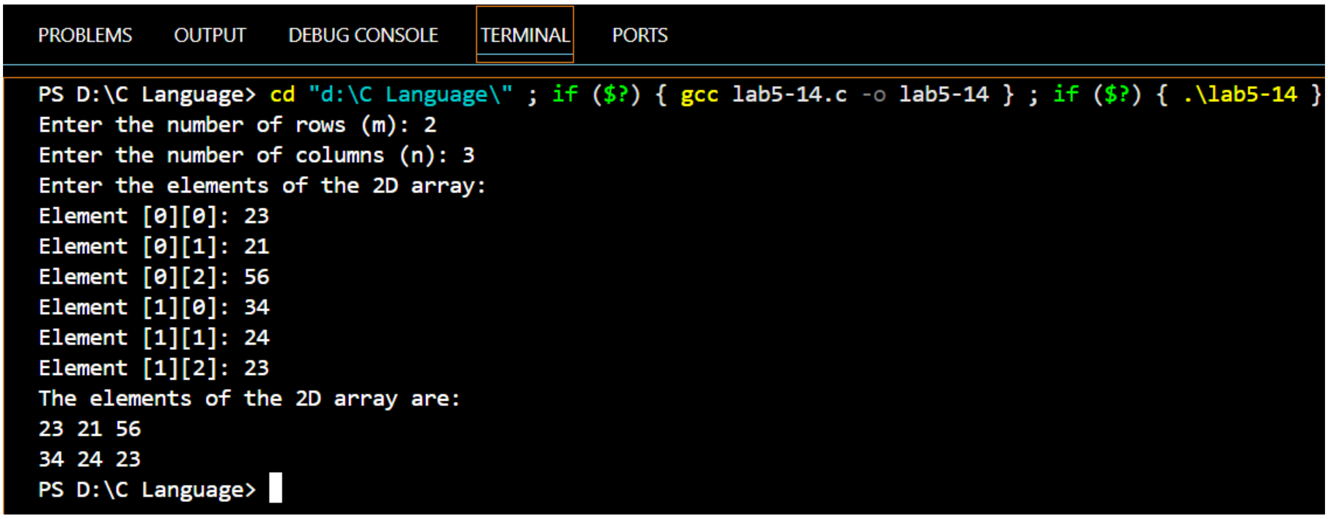
```
#include<stdio.h>

int main() {
    int n, i, j, k;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

    // Input the elements of the array
    printf("Enter %d integers:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Remove duplicate elements
    for(i = 0; i < n; i++) {
        for(j = i + 1; j < n; ) {
            if(arr[j] == arr[i]) {
                // Shift elements to the left
                for(k = j; k < n - 1; k++) {
                    arr[k] = arr[k + 1];
                }
                n--; // Reduce array size
            } else {
                j++;
            }
        }
    }

    // Print the array after removing duplicates
    printf("Array after removing duplicates:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-13.c -o lab5-13 } ; if ($?) { .\lab5-13 }
Enter the size of the array: 5
Enter 5 integers:
12
13
12
45
45
Array after removing duplicates:
12 13 45
PS D:\C Language> 
```

14. Write a program to input and display a 2D array of size m x n.

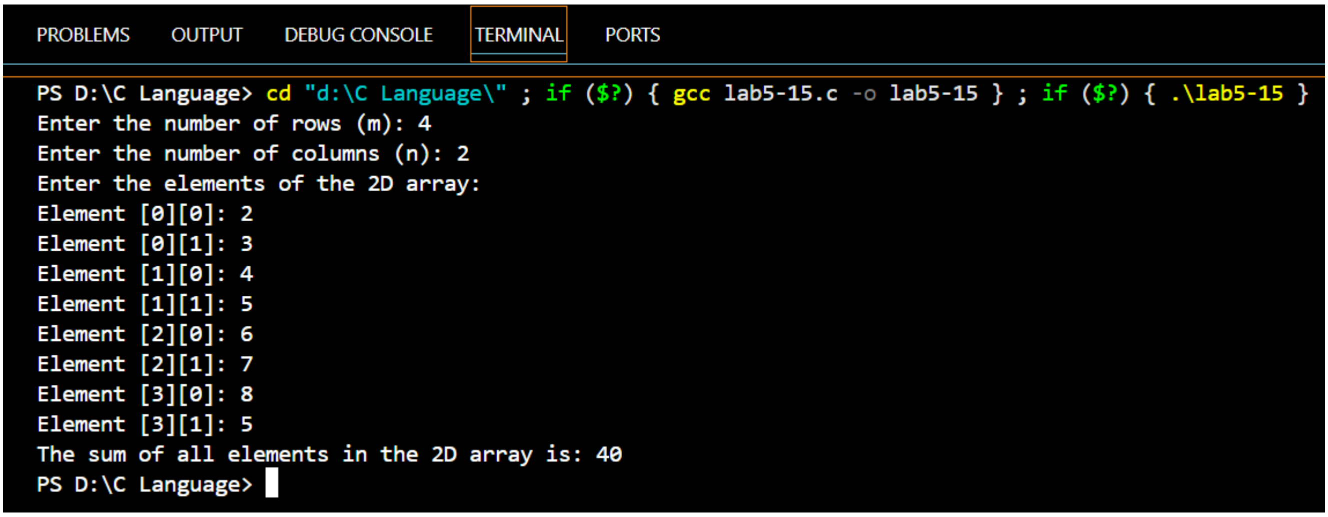
```
#include<stdio.h>

int main() {
    int m, n, i, j;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Display the elements of the 2D array
    printf("The elements of the 2D array are:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-14.c -o lab5-14 } ; if ($?) { .\lab5-14 }
Enter the number of rows (m): 2
Enter the number of columns (n): 3
Enter the elements of the 2D array:
Element [0][0]: 23
Element [0][1]: 21
Element [0][2]: 56
Element [1][0]: 34
Element [1][1]: 24
Element [1][2]: 23
The elements of the 2D array are:
23 21 56
34 24 23
PS D:\C Language>
```

15. Write a program to find the sum of all elements in a 2D array of size m x n.

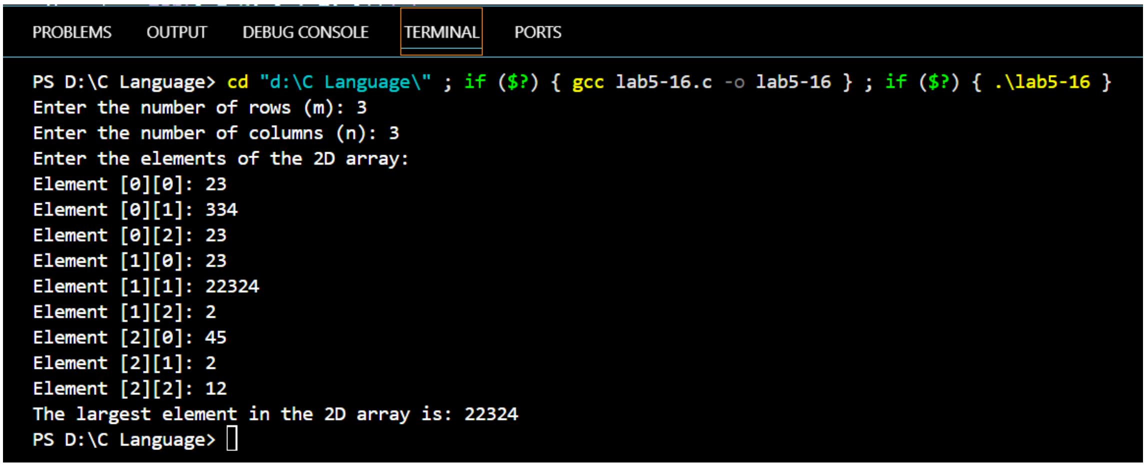
```
#include <stdio.h>

int main() {
    int m, n, i, j, sum = 0;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Calculate the sum of all elements in the 2D array
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            sum += arr[i][j];
        }
    }

    // Print the sum of all elements in the 2D array
    printf("The sum of all elements in the 2D array is: %d\n", sum);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-15.c -o lab5-15 } ; if ($?) { .\lab5-15 }
Enter the number of rows (m): 4
Enter the number of columns (n): 2
Enter the elements of the 2D array:
Element [0][0]: 2
Element [0][1]: 3
Element [1][0]: 4
Element [1][1]: 5
Element [2][0]: 6
Element [2][1]: 7
Element [3][0]: 8
Element [3][1]: 5
The sum of all elements in the 2D array is: 40
PS D:\C Language> 
```

16. Write a program to find the largest element in a 2D array of size m x n.

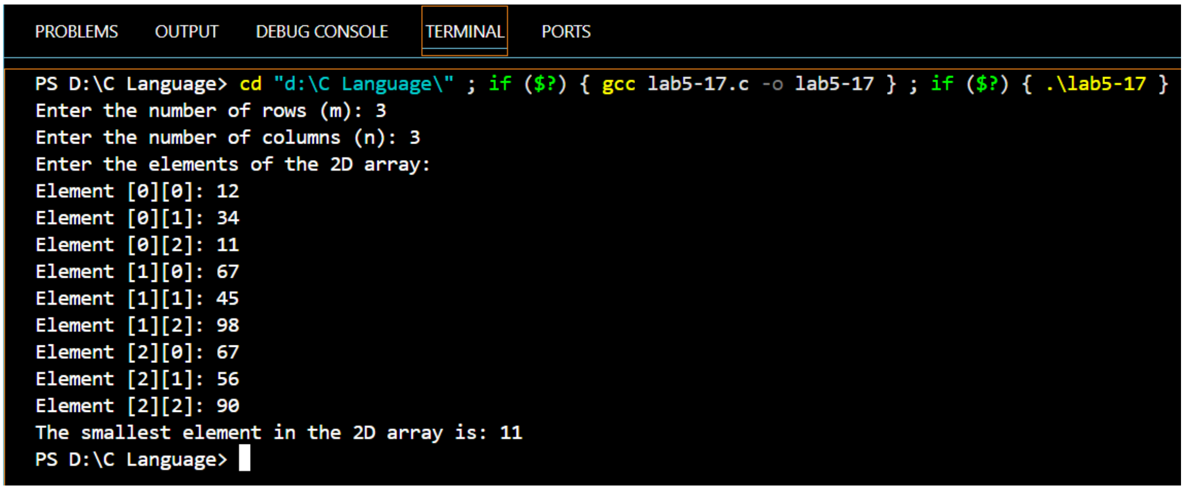
```
#include<stdio.h>

int main() {
    int m, n, i, j, max;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Assume the first element is the largest initially
    max = arr[0][0];

    // Find the largest element in the 2D array
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            if(arr[i][j] > max) {
                max = arr[i][j];
            }
        }
    }

    // Print the largest element in the 2D array
    printf("The largest element in the 2D array is: %d\n", max);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-16.c -o lab5-16 } ; if ($?) { .\lab5-16 }
Enter the number of rows (m): 3
Enter the number of columns (n): 3
Enter the elements of the 2D array:
Element [0][0]: 23
Element [0][1]: 334
Element [0][2]: 23
Element [1][0]: 23
Element [1][1]: 22324
Element [1][2]: 2
Element [2][0]: 45
Element [2][1]: 2
Element [2][2]: 12
The largest element in the 2D array is: 22324
PS D:\C Language> 
```

17. Write a program to find the smallest element in a 2D array of size m x n.

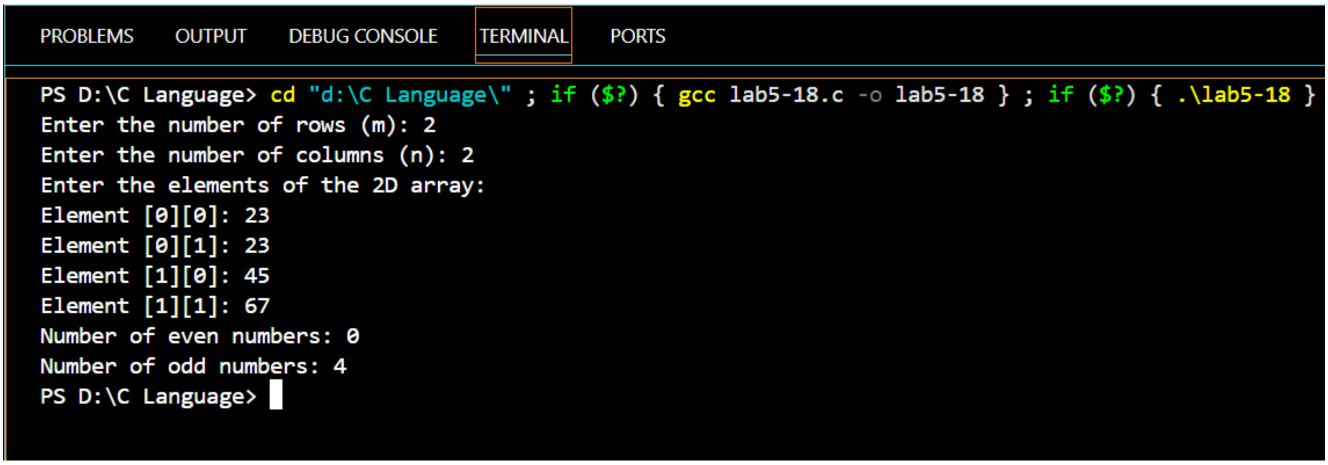
```
#include<stdio.h>

int main() {
    int m, n, i, j, min;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Assume the first element is the smallest initially
    min = arr[0][0];

    // Find the smallest element in the 2D array
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            if(arr[i][j] < min) {
                min = arr[i][j];
            }
        }
    }

    // Print the smallest element in the 2D array
    printf("The smallest element in the 2D array is: %d\n", min);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-17.c -o lab5-17 } ; if ($?) { .\lab5-17 }
Enter the number of rows (m): 3
Enter the number of columns (n): 3
Enter the elements of the 2D array:
Element [0][0]: 12
Element [0][1]: 34
Element [0][2]: 11
Element [1][0]: 67
Element [1][1]: 45
Element [1][2]: 98
Element [2][0]: 67
Element [2][1]: 56
Element [2][2]: 90
The smallest element in the 2D array is: 11
PS D:\C Language> 
```

18. Write a program to count the number of even and odd numbers in a 2D array.

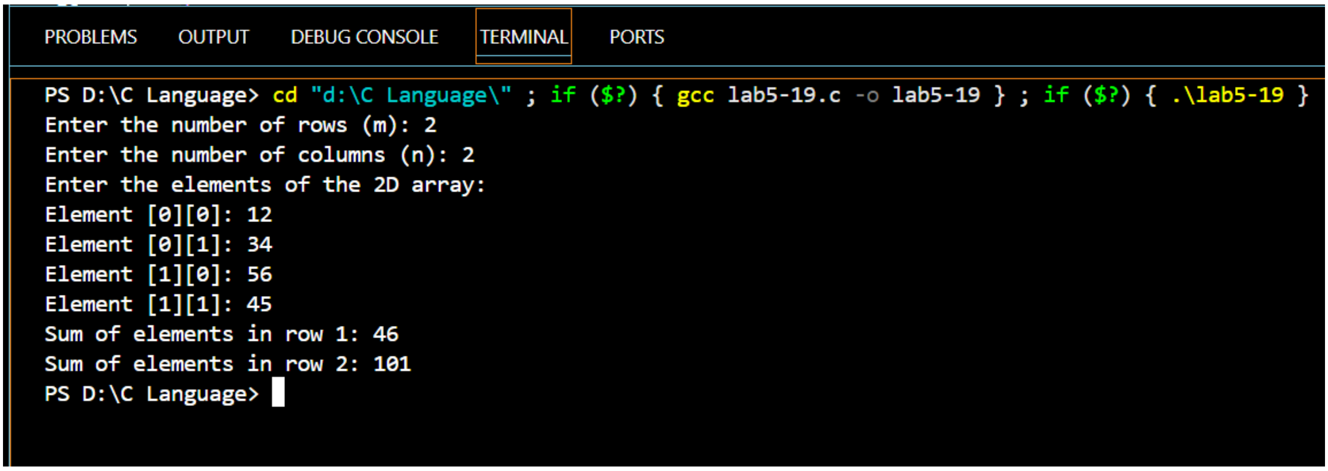
```
#include<stdio.h>

int main() {
    int m, n, i, j, evenCount = 0, oddCount = 0;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Count even and odd numbers
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            if(arr[i][j] % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
        }
    }

    // Print the counts of even and odd numbers
    printf("Number of even numbers: %d\n", evenCount);
    printf("Number of odd numbers: %d\n", oddCount);

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-18.c -o lab5-18 } ; if ($?) { .\lab5-18 }
Enter the number of rows (m): 2
Enter the number of columns (n): 2
Enter the elements of the 2D array:
Element [0][0]: 23
Element [0][1]: 23
Element [1][0]: 45
Element [1][1]: 67
Number of even numbers: 0
Number of odd numbers: 4
PS D:\C Language>
```

19. Write a program to find the sum of elements in each row of a 2D array.

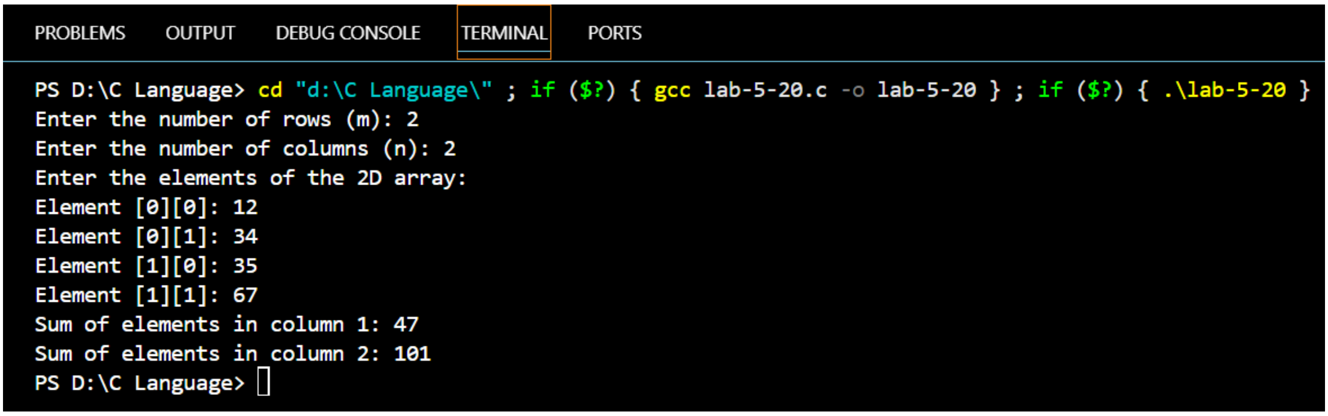
```
#include<stdio.h>

int main() {
    int m, n, i, j;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Find the sum of elements in each row
    for(i = 0; i < m; i++) {
        int rowSum = 0;
        for(j = 0; j < n; j++) {
            rowSum += arr[i][j];
        }
        printf("Sum of elements in row %d: %d\n", i + 1, rowSum);
    }

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-19.c -o lab5-19 } ; if ($?) { .\lab5-19 }
Enter the number of rows (m): 2
Enter the number of columns (n): 2
Enter the elements of the 2D array:
Element [0][0]: 12
Element [0][1]: 34
Element [1][0]: 56
Element [1][1]: 45
Sum of elements in row 1: 46
Sum of elements in row 2: 101
PS D:\C Language> █
```

20. Write a program to find the sum of elements in each column of a 2D array.

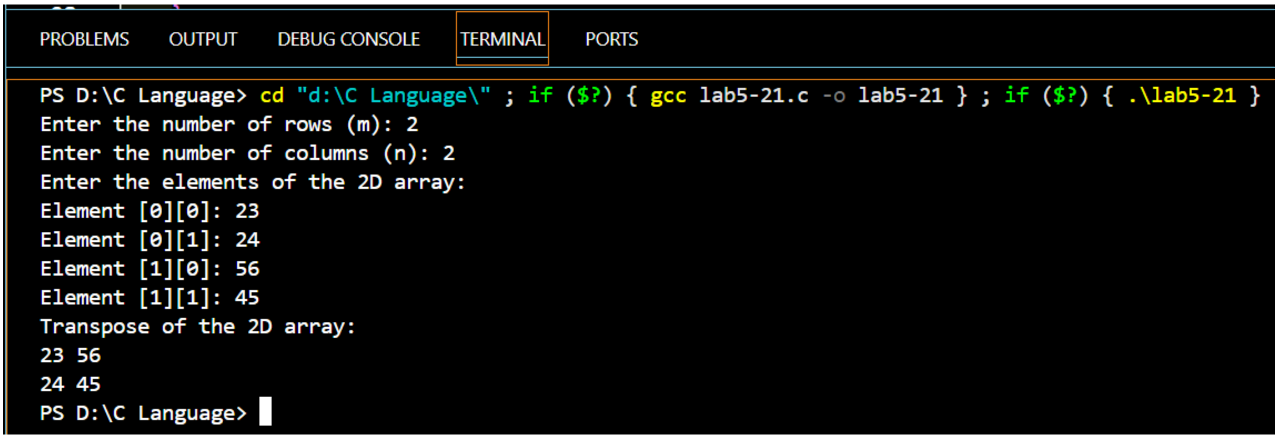
```
#include <stdio.h>

int main() {
    int m, n, i, j;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Find the sum of elements in each column
    for(j = 0; j < n; j++) {
        int colSum = 0;
        for(i = 0; i < m; i++) {
            colSum += arr[i][j];
        }
        printf("Sum of elements in column %d: %d\n", j + 1, colSum);
    }

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab-5-20.c -o lab-5-20 } ; if ($?) { .\lab-5-20 }
Enter the number of rows (m): 2
Enter the number of columns (n): 2
Enter the elements of the 2D array:
Element [0][0]: 12
Element [0][1]: 34
Element [1][0]: 35
Element [1][1]: 67
Sum of elements in column 1: 47
Sum of elements in column 2: 101
PS D:\C Language> 
```

21. Write a program to find the transpose of a 2D array (swap rows and columns).

```
#include <stdio.h>

int main() {
    int m, n, i, j;

    // Input the size of the 2D array
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int arr[m][n], transpose[n][m];

    // Input the elements of the 2D array
    printf("Enter the elements of the 2D array:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    // Find the transpose of the 2D array
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            transpose[j][i] = arr[i][j];
        }
    }

    // Print the transpose of the 2D array
    printf("Transpose of the 2D array:\n");
    for(i = 0; i < n; i++) {
        for(j = 0; j < m; j++) {
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

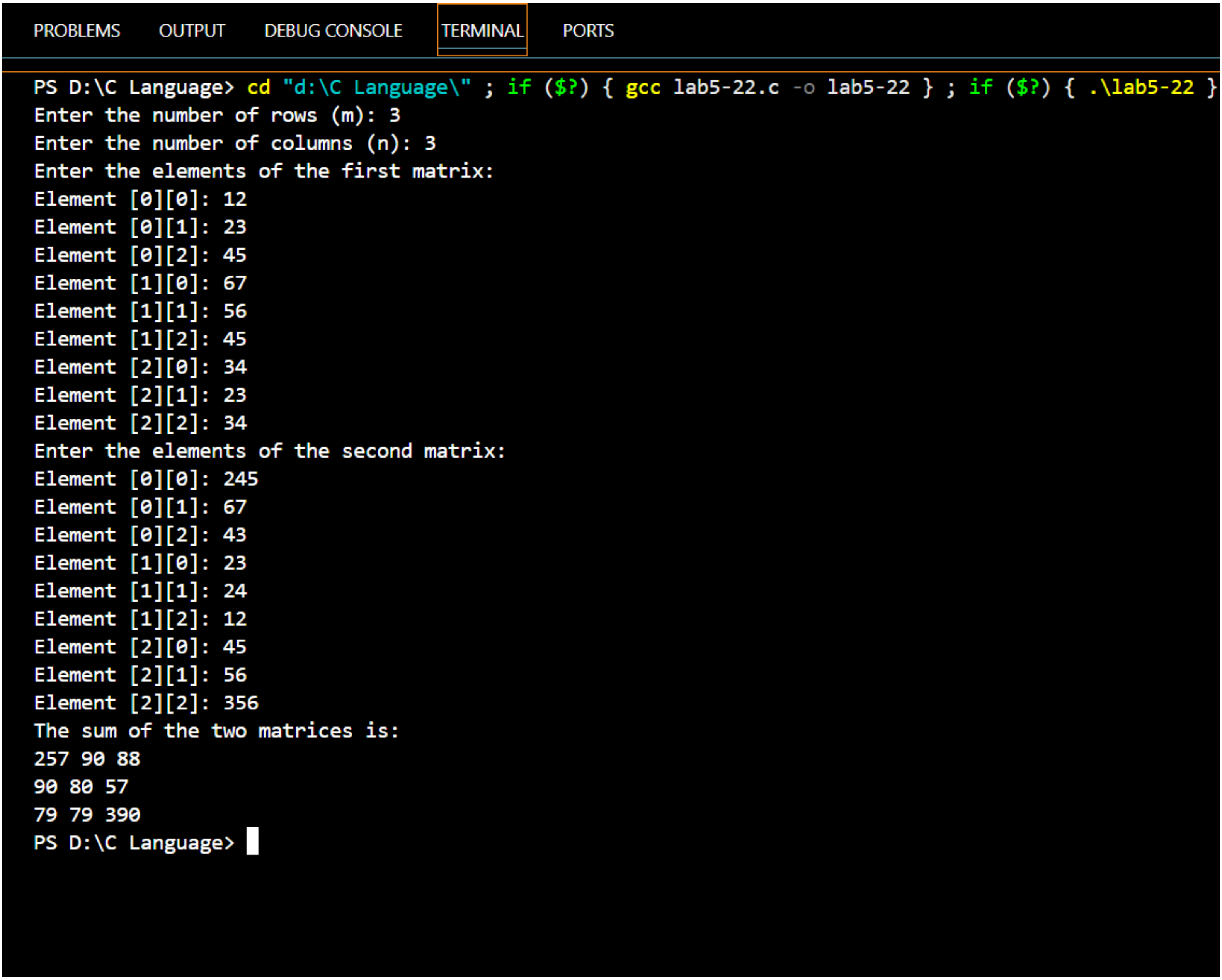


```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-21.c -o lab5-21 } ; if ($?) { .\lab5-21 }
Enter the number of rows (m): 2
Enter the number of columns (n): 2
Enter the elements of the 2D array:
Element [0][0]: 23
Element [0][1]: 24
Element [1][0]: 56
Element [1][1]: 45
Transpose of the 2D array:
23 56
24 45
PS D:\C Language>
```

22. Write a program to add two matrices.

```
#include <stdio.h>

int main() {
    int m, n, i, j;

    // Input the size of the matrices
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int mat1[m][n], mat2[m][n], sum[m][n];

    // Input the elements of the first matrix
    printf("Enter the elements of the first matrix:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &mat1[i][j]);
        }
    }

    // Input the elements of the second matrix
    printf("Enter the elements of the second matrix:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &mat2[i][j]);
        }
    }

    // Add the two matrices
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            sum[i][j] = mat1[i][j] + mat2[i][j];
        }
    }

    // Print the resulting sum matrix
    printf("The sum of the two matrices is:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

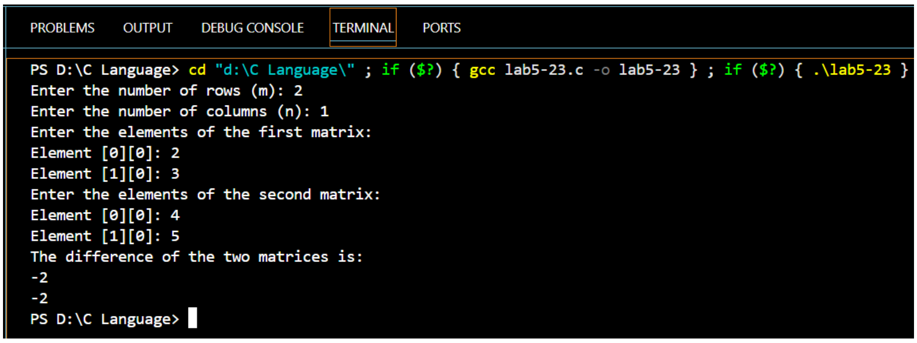
Output Of Q22:



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-22.c -o lab5-22 } ; if ($?) { .\lab5-22 }
Enter the number of rows (m): 3
Enter the number of columns (n): 3
Enter the elements of the first matrix:
Element [0][0]: 12
Element [0][1]: 23
Element [0][2]: 45
Element [1][0]: 67
Element [1][1]: 56
Element [1][2]: 45
Element [2][0]: 34
Element [2][1]: 23
Element [2][2]: 34
Enter the elements of the second matrix:
Element [0][0]: 245
Element [0][1]: 67
Element [0][2]: 43
Element [1][0]: 23
Element [1][1]: 24
Element [1][2]: 12
Element [2][0]: 45
Element [2][1]: 56
Element [2][2]: 356
The sum of the two matrices is:
257 90 88
90 80 57
79 79 390
PS D:\C Language> 
```

23. Write a program to subtract two matrices.

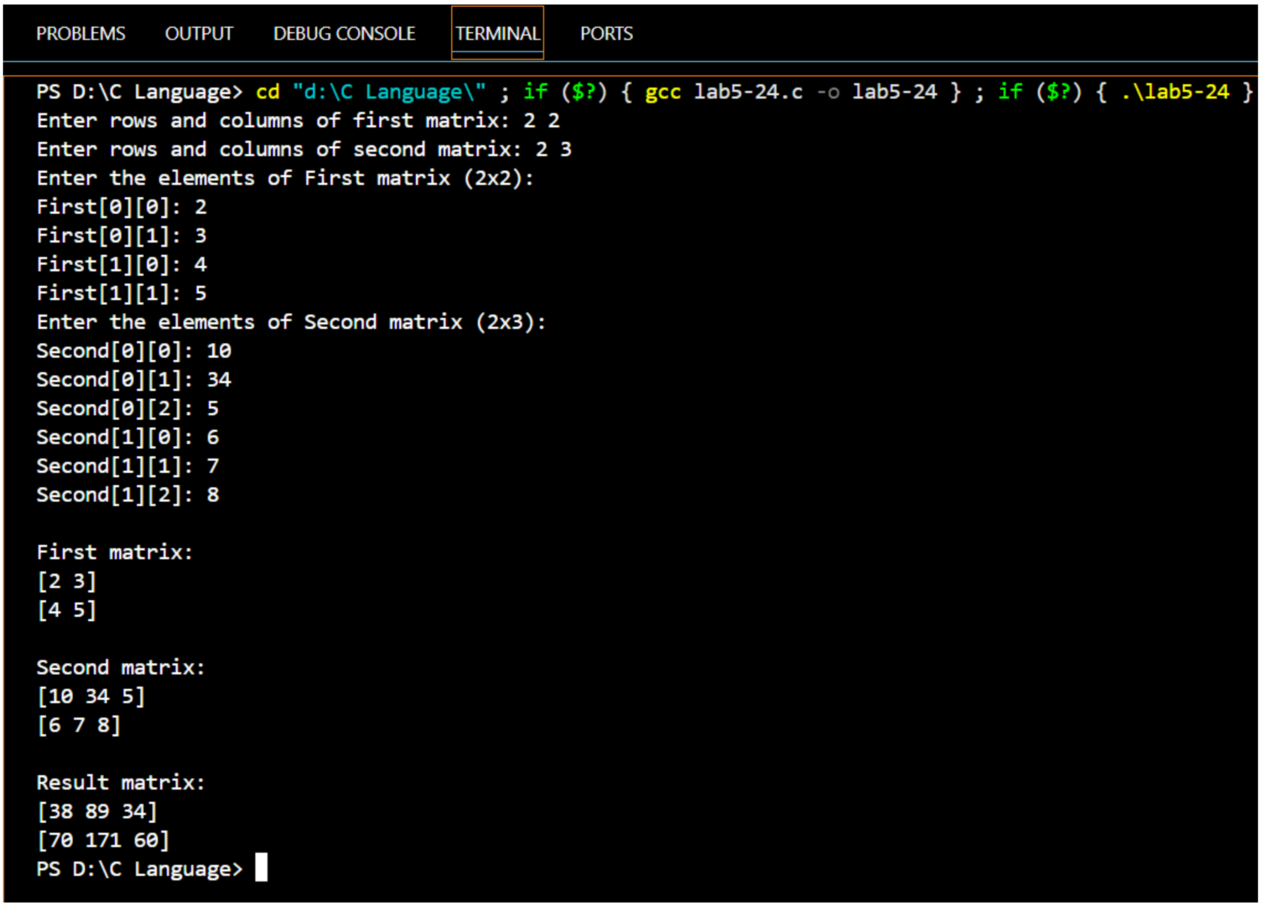
```
#include <stdio.h>

int main() {
    int m, n, i, j;

    // Input the size of the matrices
    printf("Enter the number of rows (m): ");
    scanf("%d", &m);
    printf("Enter the number of columns (n): ");
    scanf("%d", &n);

    int mat1[m][n], mat2[m][n], difference[m][n];

    // Input the elements of the first matrix
    printf("Enter the elements of the first matrix:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &mat1[i][j]);
        }
    }

    // Input the elements of the second matrix
    printf("Enter the elements of the second matrix:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &mat2[i][j]);
        }
    }

    // Subtract the two matrices
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            difference[i][j] = mat1[i][j] - mat2[i][j];
        }
    }

    // Print the resulting difference matrix
    printf("The difference of the two matrices is:\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("%d ", difference[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-23.c -o lab5-23 } ; if ($?) { .\lab5-23 }
Enter the number of rows (m): 2
Enter the number of columns (n): 1
Enter the elements of the first matrix:
Element [0][0]: 2
Element [1][0]: 3
Enter the elements of the second matrix:
Element [0][0]: 4
Element [1][0]: 5
The difference of the two matrices is:
-2
-2
PS D:\C Language> 
```

24. Write a program to add multiply two matrices.

```
#include <stdio.h>

void inputMatrix(int rows, int cols, int matrix[rows][cols], const char* name) {
    printf("Enter the elements of %s matrix (%dx%d):\n", name, rows, cols);
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%s[%d][%d]: ", name, i, j);
            scanf("%d", &matrix[i][j]);
        }
    }
}

void printMatrix(int rows, int cols, int matrix[rows][cols], const char* name) {
    printf("\n%s matrix:\n", name);
    for (int i = 0; i < rows; i++) {
        printf("[");
        for (int j = 0; j < cols; j++) {
            printf("%d", matrix[i][j]);
            if (j < cols - 1) printf(" ");
        }
        printf("]\n");
    }
}

void multiplyMatrices(int m1, int n1, int mat1[m1][n1], int m2, int n2, int mat2[m2][n2], int
result[m1][n2]) {
    for (int i = 0; i < m1; i++) {
        for (int j = 0; j < n2; j++) {
            result[i][j] = 0;
            for (int k = 0; k < n1; k++) {
                result[i][j] += mat1[i][k] * mat2[k][j];
            }
        }
    }
}

int main() {
    int m1, n1, m2, n2;

    // Input sizes
    printf("Enter rows and columns of first matrix: ");
    scanf("%d%d", &m1, &n1);
    printf("Enter rows and columns of second matrix: ");
    scanf("%d%d", &m2, &n2);

if (n1 != m2) {
        printf("Matrix multiplication is not possible. Columns of first matrix must equal rows of
second matrix.\n");
        return 1;
    }
int mat1[m1][n1], mat2[m2][n2], result[m1][n2];

    // Input elements
    inputMatrix(m1, n1, mat1, "First");
    inputMatrix(m2, n2, mat2, "Second");

    // Multiply matrices
    multiplyMatrices(m1, n1, mat1, m2, n2, mat2, result);

    // Print matrices
    printMatrix(m1, n1, mat1, "First");
    printMatrix(m2, n2, mat2, "Second");
    printMatrix(m1, n2, result, "Result");

    return 0;
}
```



```
PS D:\C Language> cd "d:\C Language\" ; if ($?) { gcc lab5-24.c -o lab5-24 } ; if ($?) { .\lab5-24 }
Enter rows and columns of first matrix: 2 2
Enter rows and columns of second matrix: 2 3
Enter the elements of First matrix (2x2):
First[0][0]: 2
First[0][1]: 3
First[1][0]: 4
First[1][1]: 5
Enter the elements of Second matrix (2x3):
Second[0][0]: 10
Second[0][1]: 34
Second[0][2]: 5
Second[1][0]: 6
Second[1][1]: 7
Second[1][2]: 8

First matrix:
[2 3]
[4 5]

Second matrix:
[10 34 5]
[6 7 8]

Result matrix:
[38 89 34]
[70 171 60]
PS D:\C Language> 
```

Conclusion

In summary, arrays are a vital data structure in C programming that allow us to store and manage collections of data efficiently. We covered:

1. **One-Dimensional Arrays (1D)**:
    a. Suitable for storing a list of variables of the same type.
    b. Useful for applications like lists, queues, and stacks.
2. **Two-Dimensional Arrays (2D)**:
    a. Ideal for representing matrices or tables.
    b. Useful for applications like spreadsheets, image processing, and game boards.

Understanding how to declare, initialize, access, and manipulate array elements is essential for efficient data handling and algorithm implementation in C programming. Arrays help organize data in a structured manner and are foundational for more complex data structures.