



## 1. TITLE OF THE LAB EXPERIMENT

Arrays (1D & 2D)

## 2. OBJECTIVES/AIM [1]

Understanding the concept of 1D and 2D Arrays from a different perspective and using this idea I have to solve real-life projects by implementing the logic.

## 3. PROCEDURE / ANALYSIS / DESIGN [2]

Since this is an advanced topic that's why I worked with Implementation and tried to get the exact output from the source code I have written.

## 4. IMPLEMENTATION [2]

### 1D Array Problems

Task-1:

```
// Take n numbers in an array and print the sum and average of the n numbers.
#include <stdio.h>

int main() {
    int n, i;
    long sum = 0;
    float average;
    int array[100];

    printf("Input the number of terms : ");
    scanf("%d", &n);

    for(i = 1; i <= n; i++){
        scanf("%d", &array[i]);
        sum += array[i];
    }
    average = sum / n;
    printf("The Sum is : %ld\n", sum);
    printf("The Average is : %.2f\n", average);

    return 0;
}
```

## Task-2:

```

// Find all prime numbers in a given array.

#include <stdio.h>

int main() {
    int n, i, j;
    int array[100];
    int flag = 0;
    printf("Input the number of terms : ");
    scanf("%d", &n);

    for(i = 1; i <= n; i++){
        scanf("%d", &array[i]);
    }

    for(i = 1; i <= n; i++){
        for(j = 2; j <= array[i]/2; j++){
            if(array[i] % j == 0){
                flag = 1;
                break;
            }
        }
        if(flag == 0) printf("%d ", array[i]);
        flag = 0;
    }
    printf("\n");

    return 0;
}
|
```

### Task-3:

```

// Find the maximum/minimum element of an array.

#include <stdio.h>

int main() {
    int a[10], i, n, max, min;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }

    max = a[0];
    min = a[0];

    for (i = 0; i < n; i++) {
        if (a[i] > max) {
            max = a[i];
        }
        if (a[i] < min) {
            min = a[i];
        }
    }
    printf("Maximum element is %d\n", max);
    printf("Minimum element is %d\n", min);

    return 0;
}
```


#### Task-4:

```
// Find the second maximum/second minimum element of an array.

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, i, j, k, l, m, max, min, temp;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
    int a[n];
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (a[i] > a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    printf("The second maximum element is: %d\n", a[1]);
    printf("The second minimum element is: %d\n", a[n - 2]);
    return 0;
}
```

**Task-5:**



```
// Write a C program to find reverse of an array.

#include <stdio.h>

int main() {
    int a[10], i, n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }
    for (i = n - 1; i >= 0; i--) {
        printf("%d ", a[i]);
    }

    return 0;
}
```

### Task-6:

```
// Write a C program to sort array elements in ascending order.

#include <stdio.h>

int main() {
    int a[10], i, n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }
    for (i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (a[i] > a[j]) {
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    for (i = 0; i < n; i++) {
        printf("%d ", a[i]);
    }

    return 0;
}
```

## 2D Array Problems

### Task-1:

```
// Write a C Program to Add Two Matrices Using Multi-dimensional Arrays.

#include <stdio.h>

int main() {
    int a[10][10], b[10][10], s[10][10], i, j, r, c;

    printf("Enter the number of rows and columns: ");
    scanf("%d %d", &r, &c);

    printf("Enter the elements of the first matrix: ");
    for (i = 0; i < r; i++) {
        for (j = 0; j < c; j++) {
            scanf("%d", &a[i][j]);
        }
    }

    printf("Enter the elements of the second matrix: ");
    for (i = 0; i < r; i++) {
        for (j = 0; j < c; j++) {
            scanf("%d", &b[i][j]);
        }
    }

    for (i = 0; i < r; i++) {
        for (j = 0; j < c; j++) {
            s[i][j] = a[i][j] + b[i][j];
        }
    }

    printf("The sum of the two matrices is: \n");
    for (i = 0; i < r; i++) {
        for (j = 0; j < c; j++) {
            printf("%d ", s[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```



## Task-2:

```
// Write a C program to Perform Matrix Multiplication.

#include <stdio.h>

int main() {
    int a[10][10], b[10][10], m[10][10], i, j, k, r1, c1, r2, c2;

    printf("Enter the number of rows and columns of the first matrix: ");
    scanf("%d %d", &r1, &c1);

    printf("Enter the elements of the first matrix: ");
    for (i = 0; i < r1; i++) {
        for (j = 0; j < c1; j++) {
            scanf("%d", &a[i][j]);
        }
    }

    printf("Enter the number of rows and columns of the second matrix: ");
    scanf("%d %d", &r2, &c2);

    if (c1 != r2) {
        printf("The matrices cannot be multiplied.");
        return 0;
    }

    printf("Enter the elements of the second matrix: ");
    for (i = 0; i < r2; i++) {
        for (j = 0; j < c2; j++) {
            scanf("%d", &b[i][j]);
        }
    }

    for (i = 0; i < r1; i++) {
        for (j = 0; j < c2; j++) {
            m[i][j] = 0;
            for (k = 0; k < c1; k++) {
                m[i][j] += a[i][k] * b[k][j];
            }
        }
    }

    printf("The product of the two matrices is: \n");
    for (i = 0; i < r1; i++) {
        for (j = 0; j < c2; j++) {
            printf("%d ", m[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

### Task-3:

```
// Write a C Program to Find Transpose of a Matrix.

#include<stdio.h>

int main(){
    int row, col, x, y;
    printf("Enter the number of row and col: ");
    scanf("%d%d", &row, &col);
    int matrix[row][col];
    int tMatrix[col][row];

    // Receiving the value of the actual matrix
    for(x = 0; x < row; x++){
        for(y = 0; y < col; y++){
            scanf("%d", &matrix[x][y]);
        }
    }

    // Transpose the matrix
    for(int x = 0; x < row; ++x){
        for(int y = 0; y < col; ++y) {
            tMatrix[y][x] = matrix[x][y];
        }
    }

    // Printing the transpose matrix
    printf("\nTranspose of the matrix:\n");
    for(x = 0; x < col; ++x){
        for(y = 0; y < row; ++y){
            printf("%d  ", tMatrix[x][y]);
            if (y == row - 1) printf("\n");
        }
    }

    return 0;
}
```

#### Task-4:

// CSE 104 course is taught in M sections and each section contains N students. Store all the student's marks in a 2D array. Print the sections with highest and lowest marks.

```
#include <stdio.h>
```

```
int main() {
    int m, n, i, j, sum = 0, max, min;
    printf("Enter the number of sections and students: ");
    scanf("%d%d", &m, &n);
    int marks[m][n];

    for (i = 0; i < m; i++) {
        printf("Enter the marks of section %d: ", i + 1);
        for (j = 0; j < n; j++) {
            scanf("%d", &marks[i][j]);
        }
    }

    // Finding the section with highest marks
    max = marks[0][0];
    min = marks[0][0];
    int maxSection = 0, minSection = 0;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            if (marks[i][j] > max) {
                max = marks[i][j];
                maxSection = i;
            }
            if (marks[i][j] < min) {
                min = marks[i][j];
                minSection = i;
            }
        }
    }

    printf("The section %d with highest marks is: %d\n", maxSection + 1, max);
    printf("The section %d with lowest marks is: %d\n", minSection + 1, min);

    return 0;
}
```

## 5. TEST RESULT / OUTPUT [2]

### 1D Array Problems

#### Task-1:

```
lab-report/Lab-4/1D Array on ☺ main [!] via C v12.1.0-gcc
➡ run
Input the number of terms : 5
1 2 3 4 5
The Sum is : 15
The Average is : 3.00
```

#### Task-2:

```
lab-report/Lab-4/1D Array on ☺ main [!] via C v12.1.0-gcc
➡ run
Input the number of terms : 10
5 10 12 29 54 23 54 45 9 11
5 29 23 11
```

#### Task-3:

```
lab-report/Lab-4/1D Array on ☺ main [!] via C v12.1.0-gcc
➡ run
Enter the number of elements: 10
1 45 565 123 3 43 21 45 65 12
Maximum element is 565
Minimum element is 1
```

#### Task-4:

```
lab-report/Lab-4/1D Array on ☺ main [!] via C v12.1.0-gcc
➡ run
Enter the number of elements in the array: 10
1 45 565 123 3 43 21 45 65 12
The second maximum element is: 3
The second minimum element is: 123
```

#### Task-5:

```
lab-report/Lab-4/1D Array on ☺ main [!] via C v12.1.0-gcc
➡ run
Enter the number of elements: 5
2 43 54 23 65
65 23 54 43 2 ↵
```

#### Task-6:

```
...eport/Lab-4/1D Array on ☺ main [!] via C v12.1.0-gcc took 21s
➡ run
Enter the number of elements: 5
12 54 76 23 5
5 12 23 54 76 ↵
```

## 2D Array Problems

#### Task-1:

```
lab-report/Lab-4/2D Array on ☺ main [x!?] via C v12.1.0-gcc
➡ run
Enter the number of rows and columns: 3 3
Enter the elements of the first matrix:
1 2 3
4 5 6
7 8 9
Enter the elements of the second matrix:
1 2 3
4 5 6
7 8 9
The sum of the two matrices is:
2 4 6
8 10 12
14 16 18
```

### Task-2:

```
lab-report/Lab-4/2D Array on ☺ main [x!?] via C v12.1.0-gcc
➡ run
Enter the number of rows and columns of the first matrix: 3 3
Enter the elements of the first matrix:
1 2 3
4 5 6
7 8 9
Enter the number of rows and columns of the second matrix: 3 3
Enter the elements of the second matrix:
1 2 3
4 5 6
7 8 9
The product of the two matrices is:
30 36 42
66 81 96
102 126 150
```

### Task-3:

```
lab-report/Lab-4/2D Array on ☺ main [x!?] via C v12.1.0-gcc
➡ run
Enter the number of row and col: 3 3
1 2 3
4 5 6
7 8 9

Transpose of the matrix:
1 4 7
2 5 8
3 6 9
```

### Task-4:

```
➡ run
Enter the number of sections and students: 3 3
Enter the marks of section 1: 1 2 3
Enter the marks of section 2: 4 5 6
Enter the marks of section 3: 7 8 9
The section 3 with highest marks is: 9
The section 1 with lowest marks is: 1
```

## **6. ANALYSIS AND DISCUSSION [2]**

1. I tested all the possible ways to solve those problems and I believe that everything is ok.
2. Yes, all the source code is working well.
3. Working with 2D Arrays is the hardest part of this assignment. Especially when working with some mathematical operation in 2D Arrays.
4. The problem set is really good. I spend a lot of time solving the last problem and a few others.
5. I like all of those problems. This makes my thought level to the next level.
6. Since I already have some previous knowledge that's why I feel comfortable with it.
7. With the skills of solving those problems from a different perspective, now I am ready for the next challenge.