LAB PROGRAM - 01

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

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

void findSecondSmallest(int arr[], int n) {
    int smallest = INT_MAX, secondSmallest = INT_MAX;
    for (int i = 0; i < n; i++) {
        if (arr[i] < smallest) {
            secondSmallest = smallest;
            smallest = arr[i];
        } else if (arr[i] < secondSmallest && arr[i] != smallest) {
            secondSmallest = arr[i];
        }
    }
    printf("Second smallest element: %d\n", secondSmallest);
}

int main() {
    int arr[] = {7, 1, 5, 3, 9};
    int n = sizeof(arr) / sizeof(arr[0]);
    findSecondSmallest(arr, n);
    return 0;
}
```

OUTPUT

```
PS C:\Users\Admin\Downloads> cd "c:\Users
Second smallest element: 3
```

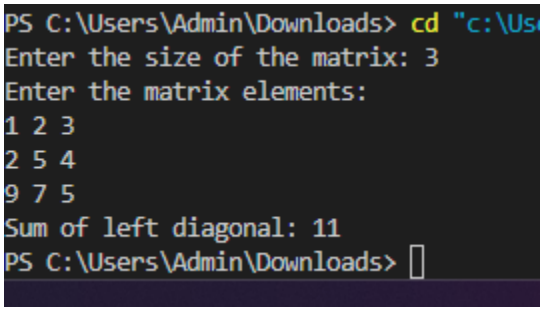
2. Write a program in C to find the sum of the left diagonals of a matrix.

```c
#include <stdio.h>

int main() {
    int n, sum = 0;
    printf("Enter the size of the matrix: ");
    scanf("%d", &n);
    int matrix[n][n];

    printf("Enter the matrix elements:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
            if (i == j) sum += matrix[i][j];
        }
    }
    printf("Sum of left diagonal: %d\n", sum);
    return 0;
}
```

OUTPUT



3.Write a program in C to find the sum of rows and columns of a matrix.

```c
#include <stdio.h>

int main() {
    int r, c;
    printf("Enter number of rows and columns: ");
    scanf("%d %d", &r, &c);
    int matrix[r][c];
    printf("Enter the matrix elements:\n");
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
            scanf("%d", &matrix[i][j]);
    printf("Sum of rows:\n");
    for (int i = 0; i < r; i++) {
        int rowSum = 0;
        for (int j = 0; j < c; j++)
            rowSum += matrix[i][j];
        printf("Row %d: %d\n", i + 1, rowSum);
    }
    printf("Sum of columns:\n");
    for (int j = 0; j < c; j++) {
        int colSum = 0;
        for (int i = 0; i < r; i++)
            colSum += matrix[i][j];
        printf("Column %d: %d\n", j + 1, colSum);
    }
    return 0;
}
```

OUTPUT

```
PS C:\Users\Admin\Downloads> cd "c:\Us
Enter number of rows and columns: 2 4
Enter the matrix elements:
1 5 9 7
3 5 7 4
Sum of rows:
Row 1: 22
Row 2: 19
Sum of columns:
Column 1: 4
Column 2: 10
Column 3: 16
Column 4: 11
PS C:\Users\Admin\Downloads>
```

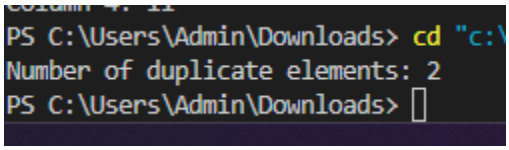
4. Write a program in C to count the total number of duplicate elements in an array.

```c
#include <stdio.h>

int countDuplicates(int arr[], int n) {
    int count = 0;
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                count++;
                break;
            }
        }
    }
    return count;
}

int main() {
    int arr[] = {2, 3, 2, 4, 5, 5, 6};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("Number of duplicate elements: %d\n", countDuplicates(arr, n));
    return 0;
}
```

OUTPUT



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

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

void findSecondLargest(int arr[], int n) {
    int largest = INT_MIN, secondLargest = INT_MIN;
    for (int i = 0; i < n; i++) {
        if (arr[i] > largest) {
            secondLargest = largest;
            largest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] != largest) {
            secondLargest = arr[i];
        }
    }
    printf("Second largest element: %d\n", secondLargest);
}

int main() {
    int arr[] = {7, 1, 5, 3, 9};
    int n = sizeof(arr) / sizeof(arr[0]);
    findSecondLargest(arr, n);
    return 0;
}
```

OUTPUT

```
PS C:\Users\Admin\Downloads> cd
Second largest element: 7
PS C:\Users\Admin\Downloads>
```

6. Write a program in C to delete an element at a desired position from an array.

```c
#include <stdio.h>

void deleteElement(int arr[], int *n, int pos) {
    for (int i = pos; i < *n - 1; i++)
        arr[i] = arr[i + 1];
    (*n)--;
}

int main() {
    int arr[10] = {1, 2, 3, 4, 5};
    int n = 5, pos;

    printf("Enter position to delete (0 to %d): ", n - 1);
    scanf("%d", &pos);

    if (pos >= 0 && pos < n) {
        deleteElement(arr, &n, pos);
        printf("Array after deletion:\n");
        for (int i = 0; i < n; i++)
            printf("%d ", arr[i]);
        printf("\n");
    } else {
        printf("Invalid position.\n");
    }
    return 0;
}
```

OUTPUT

```
PS C:\Users\Admin\Downloads> cd "c:\Us
Enter position to delete (0 to 4): 2
Array after deletion:
1 2 4 5
PS C:\Users\Admin\Downloads>
```