Lab Experiment: 01

Student Detail:

• **Name:** Prashant Joshi

• **Student ID:** 590010879

• **Branch:** MCA

• **Batch:** B1

• **Instructor:** Dr. Sourbh Kumar

Lab Assignment 1: Basic Array Operations

Problem Statement: Write a program in C that performs the following operations on an array of integers:

1. Input n elements from the user.
2. Find the largest and smallest element in the array.
3. Sort the array in ascending order.
4. Find the sum and average of the array elements.

Assignment Tasks:

- Implement an integer array of size n entered by the user.
- Perform the operations of finding the largest, smallest elements, sorting, and calculating sum and average.
- Print the array after sorting.

Solution:

```
#include <stdio.h>


void inputArray(int arr[], int n) {
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
}


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


int findSmallest(int arr[], int n) {
```

```
    int smallest = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
    }
    return smallest;
}

void sortArray(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int findSum(int arr[], int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    return sum;
}

double findAverage(int sum, int n) {
    return (double)sum / n;
}
```

```
void printArray(int arr[], int n) {
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int n;

    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    int arr[n];
    inputArray(arr, n);

    int largest = findLargest(arr, n);
    int smallest = findSmallest(arr, n);
    int sum = findSum(arr, n);
    double average = findAverage(sum, n);

    sortArray(arr, n);

    printf("Largest element: %d\n", largest);
    printf("Smallest element: %d\n", smallest);
    printf("Sum of elements: %d\n", sum);
    printf("Average of elements: %.2f\n", average);

    printArray(arr, n);

    return 0;
}
```

Output:

```
Enter the number of elements in the array: 5
Enter 5 elements:
10 25 5 40 15
```

```
Largest element: 40
Smallest element: 5
Sum of elements: 95
Average of elements: 19.00
Sorted array: 5 10 15 25 40
```

Lab Assignment 2: Array of Structures

Problem Statement: Write a program to create an array of structures to store information about n students (name, age, and marks). The program should allow the following:

1. Input details for all students.
2. Display the details of all students.
3. Sort students based on marks in descending order.
4. Find and display the student with the highest marks.

Assignment Tasks:

- Define a structure Student with fields for name, age, and marks.
- Implement functions to input, display, sort, and find the student with the highest marks.
- Display the sorted list of students based on marks.

Solution:

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

struct Student {
    char name[50];
    int age;
    float marks;
};

// Function to input details for all students
```

```
void inputStudents(struct Student students[], int n) {
    for (int i = 0; i < n; i++) {
        printf("Enter details for student %d:\n", i + 1);
        printf("Name: ");
        scanf(" %[^\n]", students[i].name);  // To read a string with spaces
        printf("Age: ");
        scanf("%d", &students[i].age);
        printf("Marks: ");
        scanf("%f", &students[i].marks);
    }
}

// Function to display details of all students
void displayStudents(struct Student students[], int n) {
    printf("\nStudent Details:\n");
    for (int i = 0; i < n; i++) {
        printf("Name: %s, Age: %d, Marks: %.2f\n", students[i].name, students[i].age, students[i].marks);
    }
}

// Function to sort students based on marks in descending order
void sortStudents(struct Student students[], int n) {
    struct Student temp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (students[i].marks < students[j].marks) {
                temp = students[i];
                students[i] = students[j];
                students[j] = temp;
            }
        }
    }
}
```

```
// Function to find and display the student with the highest marks
void displayTopStudent(struct Student students[], int n) {
    struct Student topStudent = students[0];
    for (int i = 1; i < n; i++) {
        if (students[i].marks > topStudent.marks) {
            topStudent = students[i];
        }
    }
    printf("\nStudent with the highest marks:\n");
    printf("Name: %s, Age: %d, Marks: %.2f\n", topStudent.name, topStudent.age, topStudent.marks);
}

int main() {
    int n;
    printf("Enter the number of students: ");
    scanf("%d", &n);

    struct Student students[n];
    inputStudents(students, n);

    printf("\nBefore Sorting:");
    displayStudents(students, n);

    sortStudents(students, n);

    printf("\nAfter Sorting by Marks (Descending Order):");
    displayStudents(students, n);

    displayTopStudent(students, n);

    return 0;
}
```

Output:

```
Enter the number of students: 3
Enter details for student 1:
Name: Alice
Age: 20
Marks: 85.5
Enter details for student 2:
Name: Bob
Age: 22
Marks: 90.0
Enter details for student 3:
Name: Charlie
Age: 21
Marks: 88.0
```

```
Before Sorting:
Student Details:
Name: Alice, Age: 20, Marks: 85.50
Name: Bob, Age: 22, Marks: 90.00
Name: Charlie, Age: 21, Marks: 88.00

After Sorting by Marks (Descending Order):
Student Details:
Name: Bob, Age: 22, Marks: 90.00
Name: Charlie, Age: 21, Marks: 88.00
Name: Alice, Age: 20, Marks: 85.50

Student with the highest marks:
Name: Bob, Age: 22, Marks: 90.00
```