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.

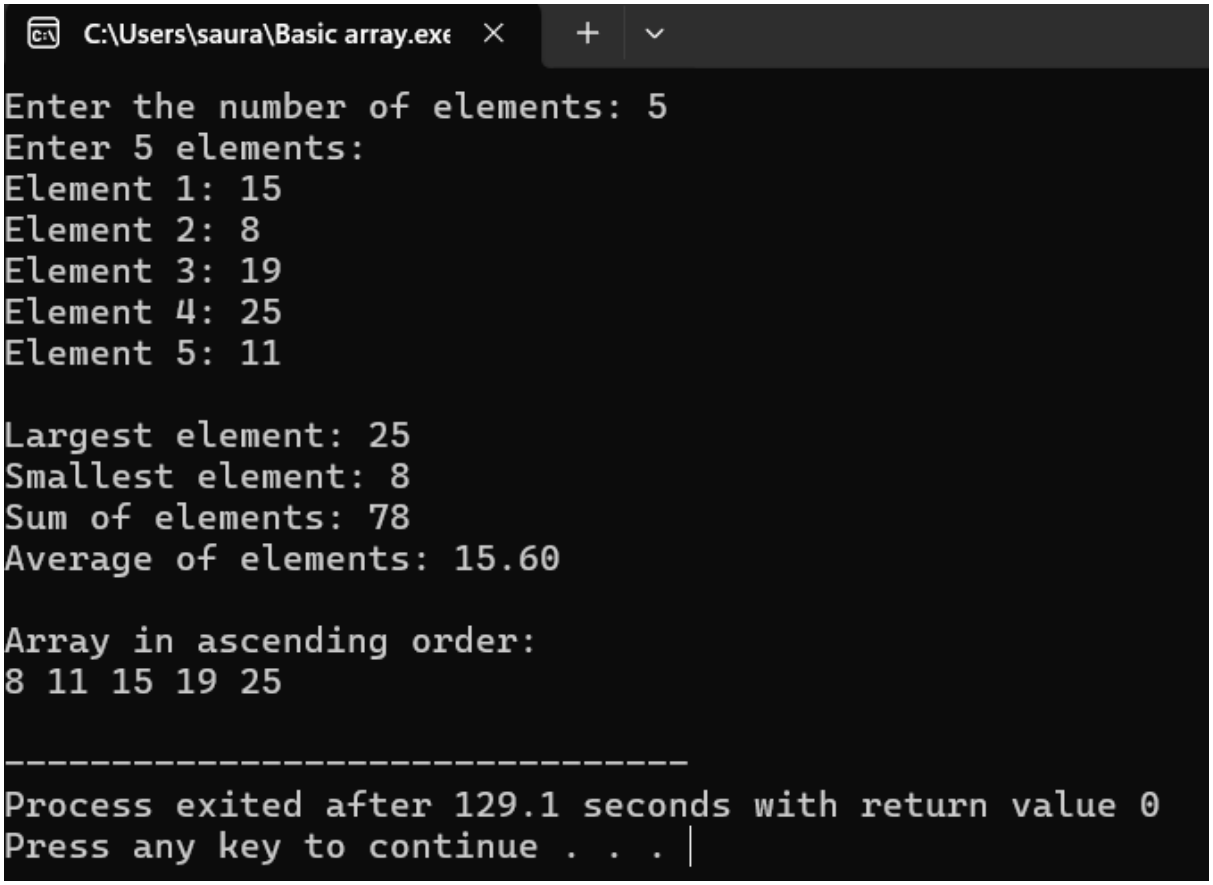
CODE:

```
#include <stdio.h>
int main() {
   int n, i, j, temp;
   int largest, smallest, sum = 0;
   float average;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   int arr[n];
   printf("Enter %d elements:\n", n);
   for (i = 0; i < n; i++) {
      printf("Element %d: ", i + 1);
      scanf("%d", &arr[i]);
      sum += arr[i];  // Calculate sum as we input elements
   }
   largest = smallest = arr[0];
   for (i = 1; i < n; i++) {
      if (arr[i] > largest) {
         largest = arr[i];
      }
      if (arr[i] < smallest) {
         smallest = arr[i];
      }}

   // Step 3: Sort the array in ascending order using Bubble Sort
   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;
         }}}
   average = (float)sum / n;
   printf("\nLargest element: %d\n", largest);
```

```
    printf("Smallest element: %d\n", smallest);
    printf("Sum of elements: %d\n", sum);
    printf("Average of elements: %.2f\n", average);
    printf("\nArray in ascending order:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
}
```

OUTPUT:



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:

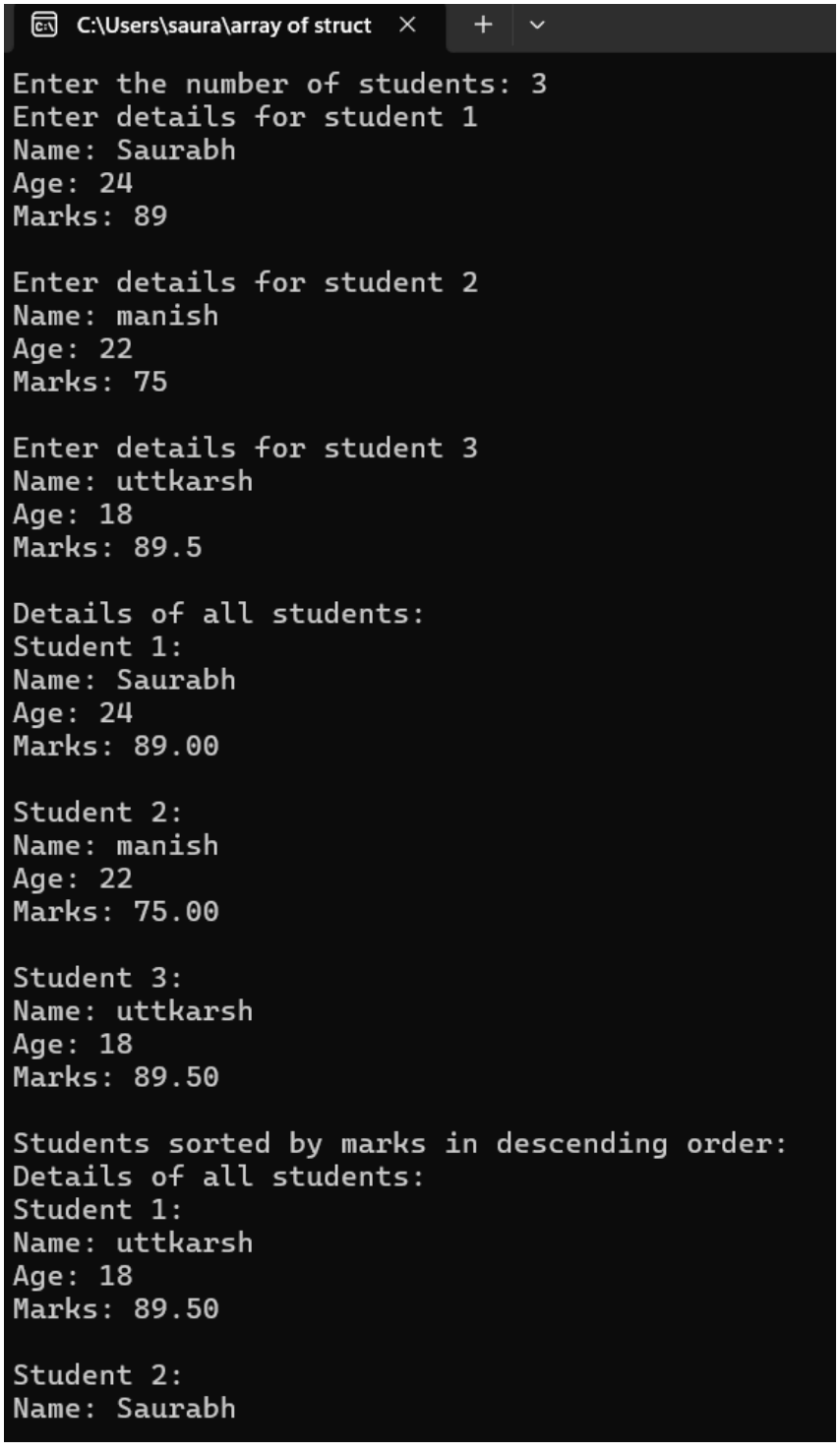
- Define a structure Student with fields for name, age, and marks.
- Implement functions to input, display, sort, and find the student with the highest marks.
- Display the sorted list of students based on marks.

CODE:

```
#include <stdio.h>
#include <string.h>
struct Student {
   char name[50];
   int age;
   float marks;
};
void inputDetails(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]s", students[i].name);
      printf("Age: ");
      scanf("%d", &students[i].age);
      printf("Marks: ");
      scanf("%f", &students[i].marks);
      printf("\n");
   }
}
void displayDetails(struct Student students[], int n) {
   printf("Details of all students:\n");
   for (int i = 0; i < n; i++) {
      printf("Student %d:\n", i + 1);
      printf("Name: %s\n", students[i].name);
      printf("Age: %d\n", students[i].age);
      printf("Marks: %.2f\n\n", students[i].marks);
   }
}
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;
            }}}}
void findHighestMarks(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("Student with the highest marks:\n");
    printf("Name: %s\n", topStudent.name);
    printf("Age: %d\n", topStudent.age);
    printf("Marks: %.2f\n", topStudent.marks);
}
int main() {
    int n;
    printf("Enter the number of students: ");
    scanf("%d", &n);
    struct Student students[n];
    inputDetails(students, n);
    displayDetails(students, n);
    // Sort students by marks in descending order and display sorted list
    sortStudents(students, n);
    printf("Students sorted by marks in descending order:\n");
    displayDetails(students, n);
    // Find and display the student with the highest marks
    findHighestMarks(students, n);
    return 0;
}
```

OUTPUT:


C:\Users\saura\array of struct
Enter the number of students: 3
Enter details for student 1
Name: Saurabh
Age: 24
Marks: 89
Enter details for student 2
Name: manish
Age: 22
Marks: 75
Enter details for student 3
Name: uttkarsh
Age: 18
Marks: 89.5
Details of all students:
Student 1:
Name: Saurabh
Age: 24
Marks: 89.00
Student 2:
Name: manish
Age: 22
Marks: 75.00
Student 3:
Name: uttkarsh
Age: 18
Marks: 89.50
Students sorted by marks in descending order:
Details of all students:
Student 1:
Name: uttkarsh
Age: 18
Marks: 89.50
Student 2:
Name: Saurabh