Experiment No. : 1

Title: Demonstrate the use of arrays, array of structure and pointers using C.

Batch: A3 **Roll No.:16010423099** **Experiment No.: 1**

Aim: Implement and demonstrate the use of arrays, array of structure and pointers using C.

Resources needed: Turbo C/C++ editor and C compiler (Online/Offline)

Theory

1) **Arrays:** An array is a collection of items stored at contiguous memory locations. It can hold multiple values of the same type.

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

2) **Structure:** A structure is a user-defined data type in C that groups different types of variables together under one name.

Example:
```
struct Person {
   char name[50];
   int age;
};
```

3) **Array of Structures:** An array of structures is a collection where each element is a structure, allowing you to store multiple records of the same type.

Example:
```
struct Person people[3];
people[0].age = 30;
strcpy(people[0].name, "Alice");
```

4) **Pointers and Pointers to Structures:** A pointer is a variable that stores the memory address of another variable. Pointers to structures allow you to access structure members using their addresses.

Example:
```
struct Person p;
struct Person *ptr = &p;
ptr->age = 25;
```

5) **Functions and Function Signature:** A function is a block of code that performs a specific task. The function signature defines the function's name, return type, and parameters.

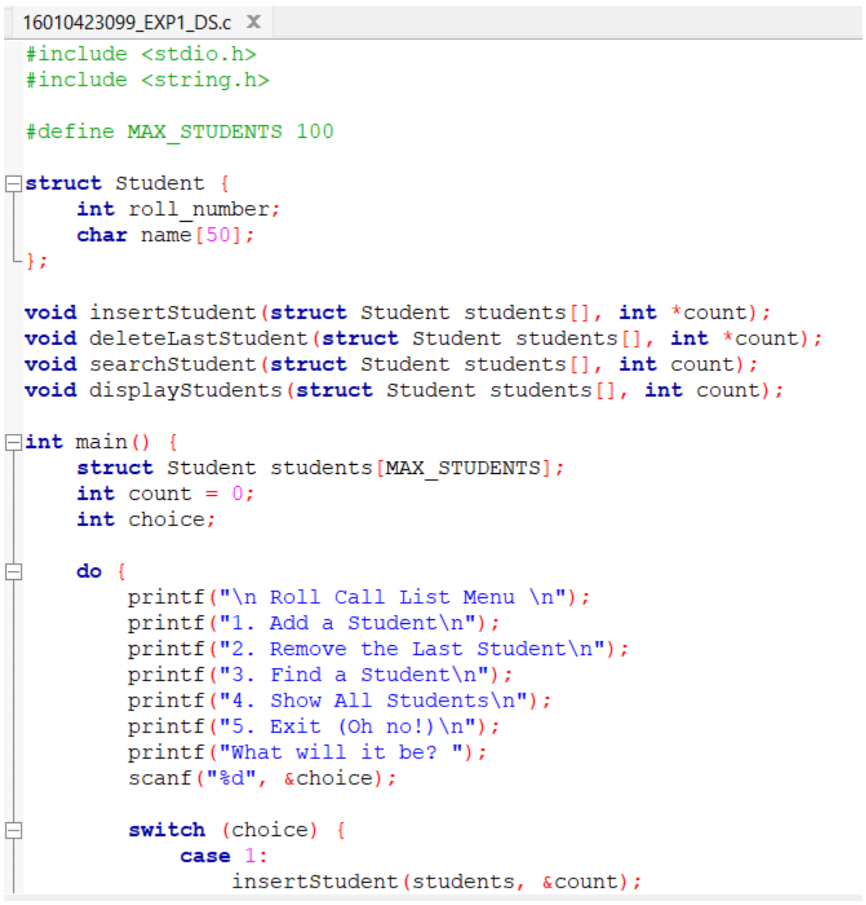
Example:
```
int add(int a, int b) {
   return a + b;
}
```

Activity : Implementing a C program to create a roll call list of a class **using an array of structure concepts**. It has the details of students as roll number and name. Program should support the following operations.

1. **Insert into the last position.**
2. **Delete from last position.**
3. **Search specific student.**
4. **Display complete list of student with details.**

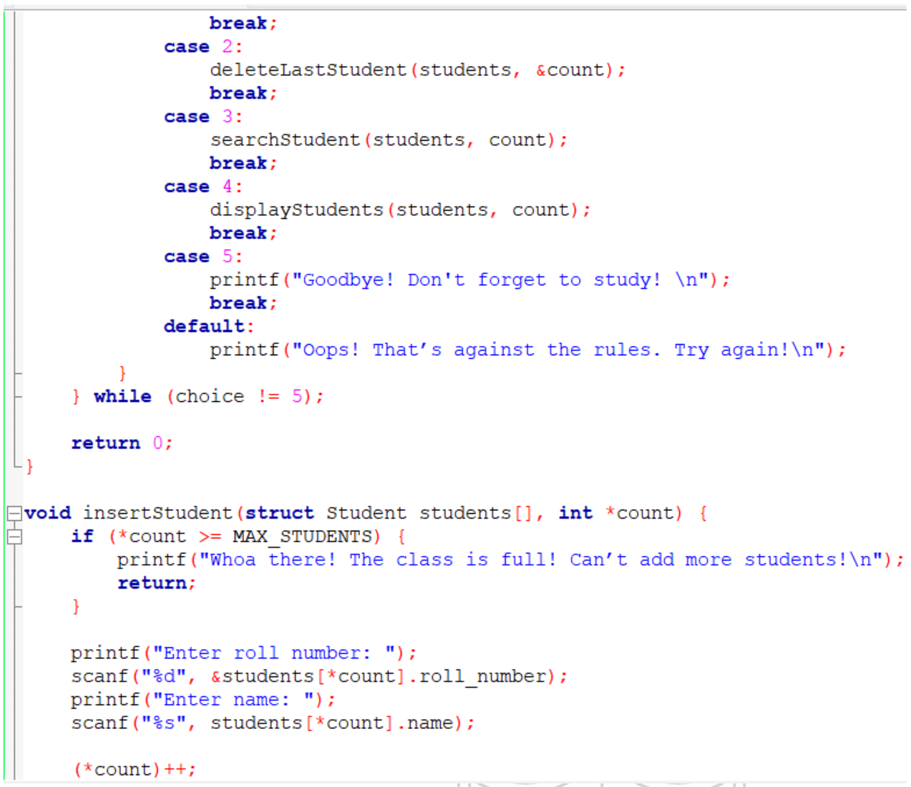
Results: A C program depicting the correct behaviour of mentioned concept and capable of handling all possible exceptional conditions/inputs and the same is reflecting clearly in the output.

Program and Output :

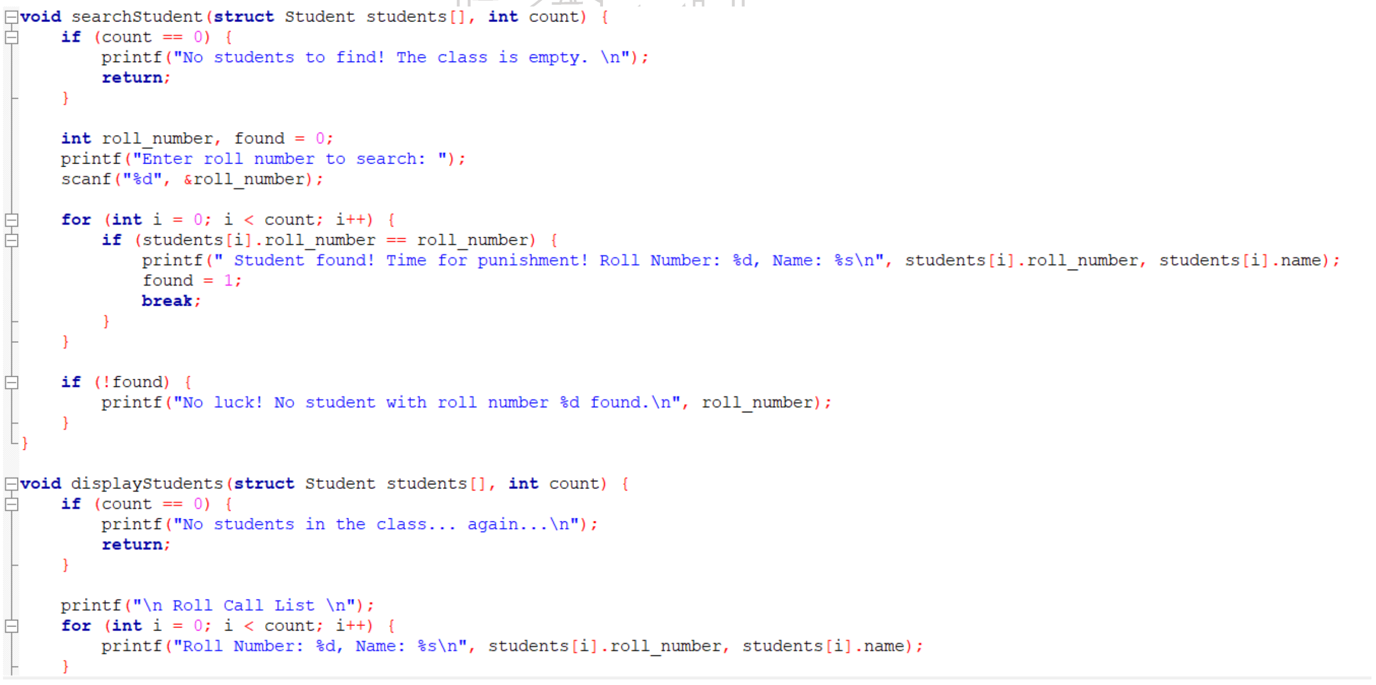


16010423099_EXP1_DS.c

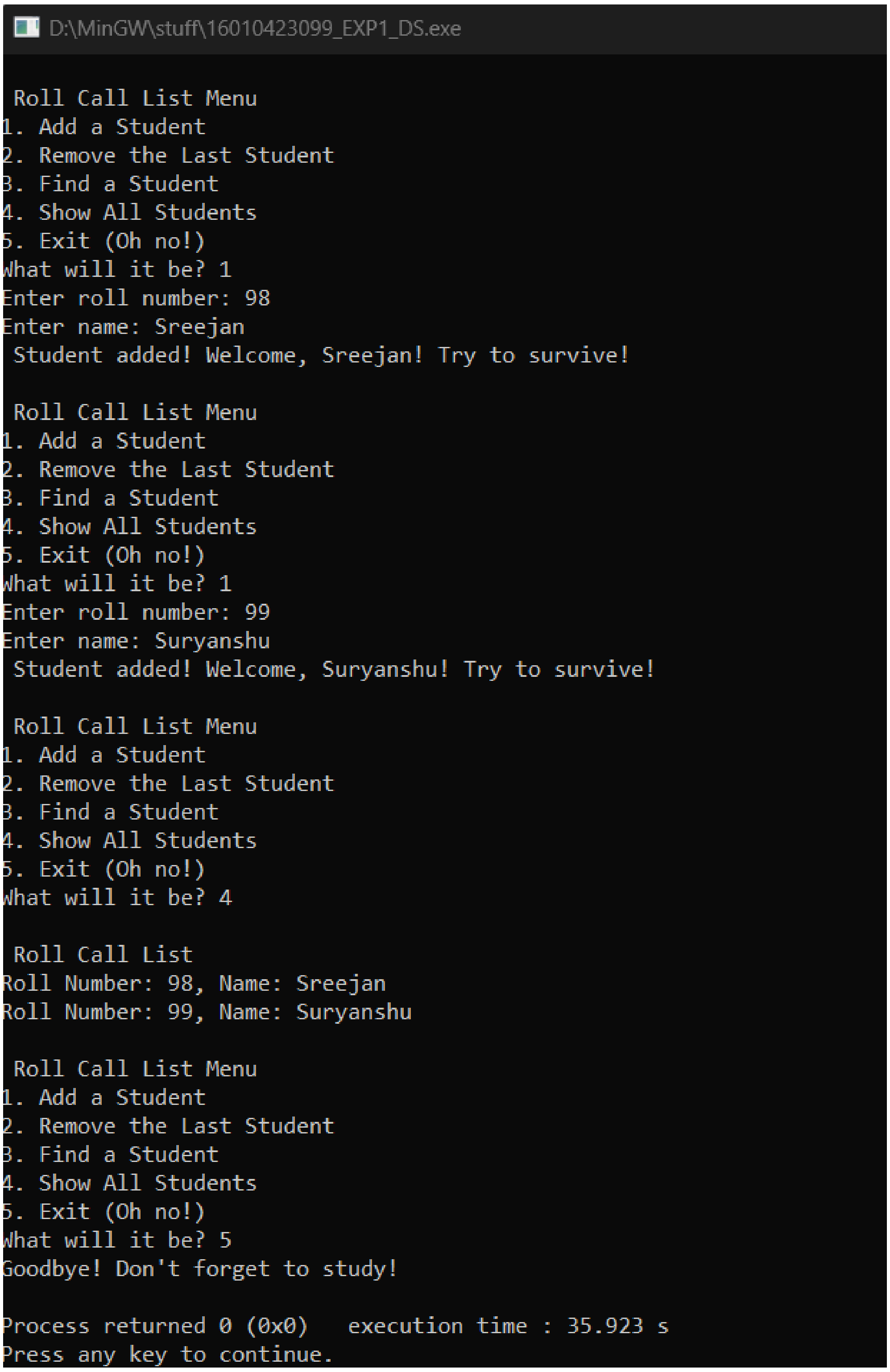
```c
#include <stdio.h>
#include <string.h>

#define MAX_STUDENTS 100

struct Student {
    int roll_number;
    char name[50];
};

void insertStudent(struct Student students[], int *count);
void deleteLastStudent(struct Student students[], int *count);
void searchStudent(struct Student students[], int count);
void displayStudents(struct Student students[], int count);

int main() {
    struct Student students[MAX_STUDENTS];
    int count = 0;
    int choice;

    do {
        printf("\n Roll Call List Menu \n");
        printf("1. Add a Student\n");
        printf("2. Remove the Last Student\n");
        printf("3. Find a Student\n");
        printf("4. Show All Students\n");
        printf("5. Exit (Oh no!)\n");
        printf("What will it be? ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                insertStudent(students, &count);
```



```c
                break;
            case 2:
                deleteLastStudent(students, &count);
                break;
            case 3:
                searchStudent(students, count);
                break;
            case 4:
                displayStudents(students, count);
                break;
            case 5:
                printf("Goodbye! Don't forget to study! \n");
                break;
            default:
                printf("Oops! That’s against the rules. Try again!\n");
        }
    } while (choice != 5);

    return 0;
}

void insertStudent(struct Student students[], int *count) {
    if (*count >= MAX_STUDENTS) {
        printf("Whoa there! The class is full! Can’t add more students!\n");
        return;
    }

    printf("Enter roll number: ");
    scanf("%d", &students[*count].roll_number);
    printf("Enter name: ");
    scanf("%s", students[*count].name);

    (*count)++;
```



```c
void searchStudent(struct Student students[], int count) {
    if (count == 0) {
        printf("No students to find! The class is empty. \n");
        return;
    }

    int roll_number, found = 0;
    printf("Enter roll number to search: ");
    scanf("%d", &roll_number);

    for (int i = 0; i < count; i++) {
        if (students[i].roll_number == roll_number) {
            printf(" Student found! Time for punishment! Roll Number: %d, Name: %s\n", students[i].roll_number, students[i].name);
            found = 1;
            break;
        }
    }

    if (!found) {
        printf("No luck! No student with roll number %d found.\n", roll_number);
    }
}

void displayStudents(struct Student students[], int count) {
    if (count == 0) {
        printf("No students in the class... again...\n");
        return;
    }

    printf("\n Roll Call List \n");
    for (int i = 0; i < count; i++) {
        printf("Roll Number: %d, Name: %s\n", students[i].roll_number, students[i].name);
    }
```



```
D:\MinGW\stuff\16010423099_EXP1_DS.exe

 Roll Call List Menu
1. Add a Student
2. Remove the Last Student
3. Find a Student
4. Show All Students
5. Exit (Oh no!)
What will it be? 1
Enter roll number: 98
Enter name: Sreejan
 Student added! Welcome, Sreejan! Try to survive!

 Roll Call List Menu
1. Add a Student
2. Remove the Last Student
3. Find a Student
4. Show All Students
5. Exit (Oh no!)
What will it be? 1
Enter roll number: 99
Enter name: Suryanshu
 Student added! Welcome, Suryanshu! Try to survive!

 Roll Call List Menu
1. Add a Student
2. Remove the Last Student
3. Find a Student
4. Show All Students
5. Exit (Oh no!)
What will it be? 4

 Roll Call List
Roll Number: 98, Name: Sreejan
Roll Number: 99, Name: Suryanshu

 Roll Call List Menu
1. Add a Student
2. Remove the Last Student
3. Find a Student
4. Show All Students
5. Exit (Oh no!)
What will it be? 5
Goodbye! Don't forget to study!

Process returned 0 (0x0)   execution time : 35.923 s
Press any key to continue.
```

Course Outcomes:

CO1. Comprehend the different data structures used in problem solving.

CO2. Apply linear and non-linear data structure in application development.

Conclusion:

Program executed successfully and knowledge of static and linear data structures like arrays, array of structure along with pointers applied.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- Y. Langsam, M. Augenstin and A. Tannenbaum, "**Data Structures using C**", Pearson Education Asia, 1st Edition, 2002
- **Data Structures A Psedocode Approach with C**, Richard F. Gilberg&Behrouz A. Forouzan, secondedition, CENGAGE Learning