DATE: LAB REPORT NO.: **7** SET: **B**

TITLE OF THE PROGRAM: **STRUCTURE**
SORT AND LIST

OBJECTIVES

I. To understand the application of structure for:
   1. Integer sorting and Alphabet (string) sorting.
   2. Working with structured data.
   3. Listing structure data on the basis of integer element and string element

REQUIREMENTS

4. C Compiler (e.g., GCC)
5. Computer System
6. IDE or Text Editor
7. OS compatible with the software

THEORY

In this project, we have taken an example program to demonstrate the objective.

Question: Create a program that takes the Roll No., First Name, Last Name, Address, and Marks of Computer Science and do the following:

1. Sort the records in descending order on the basis of Marks obtained.
2. Sort the records in ascending order on the basis of First Name.
3. Find the highest marks and count the number of students scoring the highest marks.
4. List the records of the students who failed (obtained marks is below “40”.
5. List the records of the students whose address is “Kathmandu”.

PROCEDURE (Program Code, Comment, and Output)

1. **Integer Sorting**

Program Code:

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

// Define the structure to store student information
struct Student
{
    int rN;                 // Roll Number
    char fN[30];            // First Name
    char lN[30];            // Last Name
    char a[100];            // Address
```

```
    float mC;              // Marks in Computer subject
};

int main()
{
    int n, i, j;

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

    // Declare an array of students
    struct Student s[n], temp;

    printf("Enter student details:\n");
    // Input student information from the user
    for (i = 0; i < n; i++)
    {
        printf("Student %d:\n", i + 1);
        printf("Roll Number: ");
        scanf("%d", &s[i].rN);
        printf("First Name: ");
        scanf(" %s", s[i].fN);
        printf("Last Name: ");
        scanf(" %s", s[i].lN);
        printf("Address: ");
        scanf(" %[^\n]s", s[i].a); // Allowing spaces in the address
        printf("Marks in Computer: ");
        scanf("%f", &s[i].mC);
    }

    // Sort the student records in descending order based on marks
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (s[i].mC < s[j].mC)
            {
                // Swap the student records if the current one has
lower marks
                temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }
```

```
    printf("\nStudent Records in Descending Order (Based on Marks in
Computer):\n");
    // Display the sorted student records
    for (i = 0; i < n; i++)
    {
        printf("Roll Number: %d, Name: %s %s, Address: %s, Marks in
Computer: %.2f\n", s[i].rN, s[i].fN, s[i].lN, s[i].a, s[i].mC);
    }

    return 0;
}
```

Output:

```
Enter the number of students: 3
Enter student details:
Student 1:
Roll Number: 101
First Name: Gyanendra
Last Name: Malla
Address: Kathmandu
Marks in Computer: 78.5
Student 2:
Roll Number: 102
First Name: Paras
Last Name: Khadka
Address: Kathmandu
Marks in Computer: 85.0
Student 3:
Roll Number: 103
First Name: Sompal
Last Name: Kami
Address: Lalitpur
Marks in Computer: 72.0

Student Records in Descending Order (Based on Marks in Computer):
Roll Number: 102, Name: Paras Khadka, Address: Kathmandu, Marks in Computer: 85.00
Roll Number: 101, Name: Gyanendra Malla, Address: Kathmandu, Marks in Computer: 78.50
Roll Number: 103, Name: Sompal Kami, Address: Lalitpur, Marks in Computer: 72.00
```

Explanation:

The program starts by defining a structure struct Student to store information about a student, such as roll number, first name, last name, address, and marks in the computer subject.

In the main() function, the program asks the user to enter the number of students they want to input and stores it in the variable n.

An array of student's struct Student s[n] is declared to store the student records.

The program then prompts the user to input details for each student in a loop. The loop runs n times, and in each iteration, it collects the student's roll number, first name, last name, address (allowing spaces), and marks in the computer subject.

After collecting all student details, the program sorts the student records in descending order based on the marks in the computer subject using a nested loop and a temporary variable temp for swapping.

Finally, the program prints the sorted student records in descending order (based on marks) on the console, including the roll number, full name, address, and marks in the computer subject.

2. Alphabetical (string) Sorting

Program Code:

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

// Define a structure to hold student details
struct Student
{
    int rN;          // Roll Number
    char fN[30];     // First Name
    char lN[30];     // Last Name
    char a[100];     // Address
    float mC;        // Marks in Computer
};

int main()
{
    int n, i, j;

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

    // Create an array of student structures
    struct Student s[n], temp;

    printf("Enter student details:\n");
    for (i = 0; i < n; i++)
    {
        printf("Student %d:\n", i + 1);
        printf("Roll Number: ");
        scanf("%d", &s[i].rN);
        printf("First Name: ");
        // Use " %s" to skip leading whitespace characters
        scanf(" %s", s[i].fN);
        printf("Last Name: ");
        scanf(" %s", s[i].lN);
        printf("Address: ");
        scanf(" %s", s[i].a);
        printf("Marks in Computer: ");
        scanf("%f", &s[i].mC);
```

```
    }

    // Sort student records based on first names in ascending order
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            // Compare first names using strcmp
            if (strcmp(s[i].fN, s[j].fN) > 0)
            {
                // Swap student records if they are in the wrong order
                temp = s[i];
                s[i] = s[j];
                s[j] = temp;
            }
        }
    }

    // Display sorted student records
    printf("\nStudent Records in Ascending Order (Based on First
Name):\n");
    for (i = 0; i < n; i++)
    {
        printf("Roll Number: %d, Name: %s %s, Address: %s, Marks(CS):
%.2f\n", s[i].rN, s[i].fN, s[i].lN, s[i].a, s[i].mC);
    }

    return 0;
}
```

Output:

```
Enter the number of students: 3
Enter student details:
Student 1:
Roll Number: 101
First Name: Paras
Last Name: Khadka
Address: Kathmandu
Marks in Computer: 75.5
Student 2:
Roll Number: 102
First Name: Sompal
Last Name: Kami
Address: Lalitpur
Marks in Computer: 43
Student 3:
Roll Number: 103
First Name: Gyanendra
Last Name: Malla
Address: Kathmandu
Marks in Computer: 87.5

Student Records in Ascending Order (Based on First Name):
Roll Number: 103, Name: Gyanendra Malla, Address: Kathmandu, Marks(CS): 87.50
Roll Number: 101, Name: Paras Khadka, Address: Kathmandu, Marks(CS): 75.50
Roll Number: 102, Name: Sompal Kami, Address: Lalitpur, Marks(CS): 43.00
```

Explanation:

The program begins by including the necessary header files stdio.h and string.h, which are required for input/output operations and string manipulation.

A structure struct Student is defined to hold student details, which includes the roll number (rN), first name (fN), last name (lN), address (a), and marks in Computer (mC).

In the main() function:

The user is prompted to enter the number of students (n).

An array s of struct Student is created to store the details of the students. The size of this array is determined by the user input (n).

A temporary variable temp of type struct Student is also declared to facilitate swapping of records during sorting.

The program then enters a loop to input the details of each student using a for loop. The user is asked to enter the roll number, first name, last name, address, and marks in Computer for each student.

After inputting all the student details, the program enters a nested for loop that implements the bubble sort algorithm. The sorting is done based on the students' first names in ascending order using the strcmp() function from the string.h library. If the strcmp() function returns a positive value, it means that the first name of s[i] is greater than the first name of s[j], indicating that they are in the wrong order, and a swap is performed.

The sorted student records are then displayed using another for loop. The sorted records are indicated with the roll number, full name (first name and last name), address, and marks on Computer for each student.

3. Working with Structure data.

3. List Data on the basis of Integer element

Program Code:

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

// Define the structure to store student details
struct Student
{
    int rN;             // Roll Number
    char fN[30];        // First Name
    char lN[30];        // Last Name
    char a[100];        // Address
    float mC;           // Marks in Computer
};

int main()
{
    int n, i;
```

```
    // Prompt the user to enter the number of students
    printf("Enter the number of students: ");
    scanf("%d", &n);

    // Create an array of 'n' Student structures
    struct Student s[n];

    // Prompt the user to enter student details for each student
    printf("Enter student details:\n");
    for (i = 0; i < n; i++)
    {
        printf("Student %d:\n", i + 1);
        printf("Roll Number: ");
        scanf("%d", &s[i].rN);
        printf("First Name: ");
        scanf(" %s", s[i].fN);
        printf("Last Name: ");
        scanf(" %s", s[i].lN);
        printf("Address: ");
        scanf(" %s", s[i].a);
        printf("Marks in Computer: ");
        scanf("%f", &s[i].mC);
    }

    // Display the list of students who failed in Computer Subject
    printf("\nList of Students who Failed in Computer Subject:\n");
    for (i = 0; i < n; i++)
    {
        if (s[i].mC < 40)
        {
            printf("Roll Number: %d, Name: %s %s, Address: %s, Marks
in Computer: %.2f\n",
                   s[i].rN, s[i].fN, s[i].lN,
                   s[i].a, s[i].mC);
        }
    }

    return 0;
}
```

Output:

```
Enter the number of students: 3
Enter student details:
Student 1:
Roll Number: 101
First Name: Paras
Last Name: Khadka
Address: Kathmandu
Marks in Computer: 15.5
Student 2:
Roll Number: 102
First Name: Sompal
Last Name: Kami
Address: Lalitpur
Marks in Computer: 43.0
Student 3:
Roll Number: 103
First Name: Gyanendra
Last Name: Malla
Address: Kathmandu
Marks in Computer: 17.5

List of Students who Failed in Computer Subject:
Roll No.: 101, Name: Paras Khadka, Address: Kathmandu, Marks: 15.50
Roll No.: 103, Name: Gyanendra Malla, Address: Kathmandu, Marks: 17.50
```

Explanation:

1. The program includes the necessary header files, "stdio.h" for input/output functions and "string.h" for string-related operations.
2. The structure "Student" is defined to store student details, including roll number (rN), first name (fN), last name (lN), address (a), and marks in computer subject (mC).
3. The main function starts by declaring variables 'n' (number of students) and 'i' (loop counter).
4. The user is prompted to enter the number of students.
5. An array 's' of 'n' Student structures is created to store the student details.
6. A loop is used to get the student details from the user for each student and store them in the array 's'.
7. After getting all the student details, another loop is used to display the list of students who failed in the Computer Subject (marks less than 40).

4. List Data on the basis of String element

Program Code:

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

// Define the structure to store student details
struct Student
{
    int rN;                     // Roll Number
    char fN[30];                // First Name
    char lN[30];                // Last Name
    char a[100];                // Address
    float mC;                   // Marks in Computer
};

int main()
{
    int n, i;

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

    struct Student s[n];        // Create an array of Student
structures to store student data

    printf("Enter student details:\n");
    for (i = 0; i < n; i++)
    {
        printf("Student %d:\n", i + 1);
        printf("Roll Number: ");
        scanf("%d", &s[i].rN);
        printf("First Name: ");
        scanf(" %s", s[i].fN);
        printf("Last Name: ");
        scanf(" %s", s[i].lN);
        printf("Address: ");
        scanf(" %s", s[i].a);
        printf("Marks in Computer: ");
        scanf("%f", &s[i].mC);
    }

    printf("\nList of Students whose address is Kathmandu:\n");
    for (i = 0; i < n; i++)
    {
        // Check if the address is "Kathmandu" (case-insensitive
comparison using strupr)
```

```
        if (strcmp(strupr(s[i].a), "KATHMANDU") == 0)
        {
            // Display the details of the student who lives in
Kathmandu
            printf("Roll Number: %d, Name: %s %s, Address: %s, Marks
in Computer: %.2f\n",
                   s[i].rN, s[i].fN, s[i].lN, s[i].a, s[i].mC);
        }
    }

    return 0;
}
```

Output:

```
Enter the number of students: 3
Enter student details:
Student 1:
Roll Number: 101
First Name: Paras
Last Name: Khadka
Address: Kathmandu
Marks in Computer: 75.5
Student 2:
Roll Number: 102
First Name: Sompal
Last Name: Kami
Address: Lalitpur
Marks in Computer: 43
Student 3:
Roll Number: 103
First Name: Gyanendra
Last Name: Malla
Address: Kathmandu
Marks in Computer: 87.5

List of Students whose address is Kathmandu:
Roll No.: 101, Name: Paras Khadka, Address: Kathmandu, Marks: 75.50
Roll No.: 103, Name: Gyanendra Malla, Address: Kathmandu, Marks: 87.50
```

Explanation:

1. The program starts by including the necessary header files: <stdio.h> for input/output functions and <string.h> for string manipulation functions.
2. The structure struct Student is defined, containing the following members:
    - rN: An integer to store the roll number of the student.
    - fN: A character array of size 30 to store the first name of the student.

- lN: A character array of size 30 to store the last name of the student.
- a: A character array of size 100 to store the address of the student.
- mC: A floating-point variable to store the marks in Computer of the student.

3. In the main() function, variables n (number of students) and i (loop counter) are declared.
4. The user is prompted to enter the number of students (n) using printf() and scanf() functions.
5. An array s of type struct Student is declared with size n, to store the details of all students.
6. The program then prompts the user to enter the details of each student using a loop. Inside the loop, it asks for the roll number, first name, last name, address, and marks in Computer for each student.
7. After taking input for all students, the program proceeds to find and display the details of students whose address is "Kathmandu". It uses a loop to iterate through the array of student structures.
8. Inside the loop, the strupr() function is used to convert the address of each student to uppercase (case-insensitive comparison), and then strcmp() is used to check if the address is "KATHMANDU" (uppercase). If the condition is true, the student's details (roll number, name, address, and marks) are displayed using printf().
9. The program ends by returning 0 from the main() function.

CONCLUSION

In this project, provided insights into the practical applications of structures, including integer and string sorting, and working with structured data. The knowledge gained enhances our ability to organize and manage data efficiently, laying the groundwork for tackling more complex challenges and optimizing future projects.