NATIONAL INSTITUTE OF TECHNOLOGY SIKKIM
Department of Computer Science and Engineering

CS13201 Data Structure and Algorithms Laboratory

Odd Semester, July–December 2024

Laboratory Assignment #1

NB: Solve the following programming problems according to instructions given during the laboratory sessions. Along with the requirements mentioned in the text of the questions, additional instructions may be given by the course instructor to produce the desired output. All programs should be written in the C programming language. Do not use the *string.h* header.

1. Write a program to store a set of names of students in an array and then find out the length of each name. Take the number of students and the names of the students as input from the user. After taking all inputs, print the length values as the final output of your program.

2. Write a program to take the names and roll numbers of *n* students as input from the user (taking *n* also as input from the user), and then ask the user to enter a roll number so that your program fetches the corresponding name of the student. First create the database of the students, and then accept the query from the user. The result of the query, i.e. the name of the student queried for, will be the final output of the program. Do not use any structures.

3. Write a program to declare a structure, with the tag *student*, which contains three variables: i) *name*, which can store a string, ii) *marks*, which can store a fraction, and iii) *length*, which can store an integer. Take the values of *name* and *marks* from the user, and then calculate the length of *name* and store it in *length*. Display all three values as the final output of your program.

4. Write a program to declare an array of structures, where each element of the array is a structure containing three variables, *name*, *marks*, and *length*, similar to Question no. 3 above. Take the names and marks of a class of *n* students as input from the user (taking *n* also as input from the user), and then store these *n* sets of data in the array of structures declared earlier, using the variable *name* to store names and the variable *marks* to store marks. Calculate and store the length of each name in the variable *length*. Display the following as the final output of your program: i) the longest name, ii) the shortest name, iii) the highest marks along with the corresponing student's name, iv) the lowest marks along with the corresponding student's name, v) the average marks of the whole class, and vi) the number of students who have passed if pass marks is 40.

5. Solve the same problem given in Question no. 2 above, but by using an array of structures where each instance of the structure will contain the name and roll number of a student. As in Question no. 2, the name of the student queried for will be the final output of the program.

6. Write a program to take an array of strings as input from the user, and then modify the array such that all the strings in the array are reversed, without using any second array, i.e. you may use only one two-dimensional array in your program. Display the modified array of strings as the final output of your program. Reversing the string means to rearrange the characters of a string in the reverse order compared to the original string, i.e. reversing the string "computer" gives the resultant string as "retupmoc".

7. Write a program to enter a matrix from the user where each element of the matrix is an integer, and then print the sum of all elements of the matrix.

8. Write a program to declare an array of structures where each instance of the structure stores the name of an employee, the salary of the employee, and the age of the employee. Display as the final output of

your program the list of names of employees who are above 30 years of age and receive a salary greater than Rs. 50,000.

9. Write a program to take an array of integers as input from the user, and then calculate the sum of all the odd numbers of the array using a recursive function called *add_odds*. There should be no iterative statement in *add_odds* to calculate the required sum. Display the required sum as the final output of your program.

10. Write a program to declare a structure to represent a rectangle, and then find out the largest rectangle from a set of rectangles. Declare an array of structures where every instance of the array is a structure containing the length and breadth of a rectangle, where the length and breadth of *n* rectangles are taken as input from the user and stored in this array of structures (*n* is also taken as user input). Display the index number of the rectangle with the largest area as the final output of your program, i.e. if the m^{th} entered rectangle is the largest, *m* should be the final output.

----------x----------