**1.Files and Structure:**

Problem Statement:

File: student.txt

3

01 Pritha 50

02 Anurina 70

03 Aradhita 30

A set of students information roll, name, marks are given in a file. Read the name of the file through command line arguments. Allocate an array of student records (structure) using dynamic memory allocation. Print the list of students along with their marks (one per line) in increasing order and write their ranks: ie. rank, roll, name, marks.

• Input example :

(Inside the file)

3

01 Pritha 50

02 Anurina 70

03 Aradhita 30

• Output example :

1 : 2 Anurina 70

2 : 1 Pritha 50

3 : 3 Aradhita 30

Proposed C Code:

/\* ------- main.c ------- \*/

#include <stdio.h>

#include <stdlib.h>

typedef struct

{

    int roll, marks;

    char name[100];

} student;

*// Sorting the array of structure*

void sort(student s1[], int n)

{

    for (int i = 0; i < n; i++)

    {

        for (int j = i + 1; j < n; j++)

        {

            if (s1[i].marks < s1[j].marks)

            {

                student temp = s1[i];

                s1[i] = s1[j];

                s1[j] = temp;

            }

        }

    }

    return;

}

int main(int argc, char \*argv[])

{

    student \*s1;

    student \*s2;

    FILE \*fpr, \*fwr;

    fpr = fopen(argv[1], "r"); *// reading from file*

    fwr = fopen(argv[2], "w"); *// writing from file*

    if (fpr == NULL)

    {

        printf("Not Opened");

    }

    int n;

    fscanf(fpr, "%d", &n);

    s1 = (student \*)malloc(n \* sizeof(student));

    s2 = (student \*)malloc(n \* sizeof(student));

    for (int i = 0; i < n; i++)

    {

        fscanf(fpr, "%d%s%d", &s1[i].roll, s1[i].name, &s1[i].marks);

    }

    sort(s1, n);

    for (int i = 0; i < n; i++)

    {

        fprintf(fwr, "%d : %d %s %d\n", i + 1, s1[i].roll, s1[i].name, s1[i].marks);

        printf("%d : %d %s %d\n", i + 1, s1[i].roll, s1[i].name, s1[i].marks);

    }

    fclose(fpr); *//Closing the files*

    fclose(fwr);

    return 0;

}

/\* ---------------------- \*/

Conclusion:

The proposed algorithm has a runtime of O(n), where n is the number of lines in student.txt file.

Limitations and assumptions for this algorithm include:

1. For this program “student.txt” must be present.
2. In command line we have to write:

gcc filename // for compilation

./execution\_file\_name reading\_file\_location writing\_file \_location

1. File location should be given in “E:\\collage\\c\\File” in that way.