**7. Topic – File Handling in C with Structures**

* **Problem Statement**

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:** /\* Here user will mention the whole directory path of the file student.txt as a command line argument in command prompt\*/

File : student.txt

6

01 Varun 92

02 Ankan 91

03 Ankur 99

04 Gourav 90

05 saptajit 96

06 Tunir 98

**Output example:**

The total number of students are 6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*After sorting of the students based on their marks in increasing order\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rank Marks Roll Name

6 90 4 Gourav

5 91 2 Ankan

4 92 1 Varun

3 96 5 saptajit

2 98 6 Tunir

1 99 3 Ankur

* **Proposed C Code**

**/\* ---------- fileprogram.c--------------- \*/**

#include <stdio.h>

#include<stdlib.h>

#include<string.h>

struct student

{

int marks,roll;

char name[25];

};

typedef struct student Std;

/\* A structure template of student is generated which contains the details of roll,marks,name as of text file \*/

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

/\* Here command line argument is used to pass commands to main() \*/

{

FILE\*ptr=NULL;

/\* FILE pointer is initialized to NULL \*/

ptr = fopen(argv[1],"r");

/\* Here we should give the second command as the directory path of student.txt file to the main() function \*/

if(ptr==NULL)

{

printf("The file cannot be opened\n");

exit(0);

/\* If there is no file then we terminate the program by giving exit(0) \*/

}

int number= fgetc(ptr)-'0';

/\* The first number in the text file is the number of students and we create an array of structure of that number size \*/

printf("The total number of students are %d\n",number);

Std\* stu = (Std\*)malloc(number\*sizeof(Std));

for ( int i = 0 ; i < number ; i++ )

{

/\* Storing the details from the file to the structure array \*/

fscanf(ptr,"%d",&stu[i].roll);

fscanf(ptr,"%s",&stu[i].name);

fscanf(ptr,"%d",&stu[i].marks);

}

/\* Variables needed for insertion sort are declared below \*/

int sort\_marks,shift\_roll;

char shift\_name[20];

for ( int i = 1 ; i < number ; i++ )

{

sort\_marks = stu[i].marks;

shift\_roll = stu[i].roll;

strcpy(shift\_name,stu[i].name);

int j;

for ( j = i - 1 ; ( j >= 0 ) && ( stu[j].marks > sort\_marks ) ; j-- )

{

stu[j+1].marks = stu[j].marks;

stu[j+1].roll = stu[j].roll;

strcpy(stu[j+1].name,stu[j].name);

}

stu[j+1].marks = sort\_marks;

stu[j+1].roll = shift\_roll;

strcpy(stu[j+1].name,shift\_name);

/\* The marks are sorted by insertion sort algorithm and name,roll are sorted marks wise. \*/

}

/\* The description of the students with rank,roll,name according to increasing marks are printed in a tabular format\*/

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*After sorting of the students based on their marks in increasing order\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("Rank\t Marks\t Roll\t Name\n\n");

for ( int i = 0 ; i < number ; i++ )

{

printf("%2d\t %3d\t %2d\t %s\n",number - i,stu[i].marks,stu[i].roll,stu[i].name);

}

return 0;

}

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

* **Conclusion**

**The proposed algorithm has a runtime of O(n^2) as we are using insertion sort where n is the size of the structure array under consideration.**

* **Limitations : This program will take commands from the user as per the question. If the source code and the text file are not in same folder then we have to mention the whole path of directories i.e the whole location of the text file as the second command or argv[1] after mentioning the .exe file while running the program from the program’s current directory in command prompt.**
* **Assumptions: We are considering the names of the students as a string without any space i.e only the first names .**