**ASSIGNMENT 3 (DBMS)**

U20CS110

KRISHNA PANDEY

1) CODE:

#include <stdio.h> #include <stdlib.h> #include <string.h> #include <time.h>

*// student info*

**struct** student

{

**int** recno;

**int** idd;

**char** fnn[20]; **char** lnn[20]; **char** gn[5]; **char** brn[4]; **char** city[15]; **int** age;

**int** jmarks;

} stu[40];

*//operator*

**int** dsc(**const void \***a, **const void \***b)

{

**struct** student **\***x = (**struct** student **\***)a; **struct** student **\***y = (**struct** student **\***)b; return -(x->jmarks - y->jmarks);

}

*// creating file*

**void** create(**int** n)

{

FILE \*fptr;

fptr = fopen("student.txt", "w"); if (fptr != NULL)

{

printf("File created succesfully.\n");

**int** j;

for (j = 0; j < n; j++)

{

printf("Enter id: "); scanf("%d", &stu[j].idd); printf("Enter first name: ");

scanf("%s", stu[j].fnn);

printf("Enter last name: "); scanf("%s", stu[j].lnn); printf("Enter gender: "); scanf("%s", stu[j].gn); printf("Enter branch: "); scanf("%s", stu[j].brn); printf("Enter city: "); scanf("%s", stu[j].city); printf("Enter age: "); scanf("%d", &stu[j].age); printf("Enter jee marks: "); scanf("%d", &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[0]), dsc); for (j = 0; j < n; j++)

{

fprintf(fptr, "%02d %d %20s %20s %5s %4s %15s %d %d\n", j + 1, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

memset(stu[j].gn, 0, 5);

memset(stu[j].brn, 0, 4);

memset(stu[j].fnn, 0, 20);

memset(stu[j].lnn, 0, 20);

memset(stu[j].city, 0, 15); fclose(fptr);

}

else

printf("File not found.\n");

}

*// counting no of records*

**int** no\_of\_rec()

{

FILE \*fptr; **int** cl = 0; **char** chr;

fptr = fopen("student.txt", "r"); chr = getc(fptr);

while (chr != EOF)

{

if (chr == '\n')

{

cl++;

}

chr = getc(fptr);

}

fclose(fptr); return cl;

}

*// display the records*

**void** display()

{

**int** n, j, i;

n = no\_of\_rec(); FILE \*fptr;

fptr = fopen("Student.txt", "r"); if (fptr != NULL)

{

printf("Student Details:-- \n"); for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", stu[j].recno, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

memset(stu[j].gn, 0, 5);

memset(stu[j].brn, 0, 4);

memset(stu[j].fnn, 0, 20);

memset(stu[j].lnn, 0, 20);

memset(stu[j].city, 0, 15);

}

fclose(fptr);

}

else

{

printf("File not found.\n");

}

}

*// add a new record*

**void** add()

{

**int** n, j, i;

n = no\_of\_rec(); FILE \*fptr;

fptr = fopen("student.txt", "r"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

fclose(fptr);

printf("File found succesfully.\n"); printf("Enter id: ");

scanf("%d", &stu[n].idd); printf("Enter first name: "); scanf("%s", stu[n].fnn); printf("Enter last name: "); scanf("%s", stu[n].lnn); printf("Enter gender: "); scanf("%s", stu[n].gn); printf("Enter branch: "); scanf("%s", stu[n].brn); printf("Enter city: "); scanf("%s", stu[n].city); printf("Enter age: "); scanf("%d", &stu[n].age); printf("Enter jee marks: "); scanf("%d", &stu[n].jmarks);

qsort(stu, n + 1, sizeof(stu[n]), dsc); fptr = fopen("student.txt", "w");

for (j = 0; j < n + 1; j++)

{

fprintf(fptr, "%02d %d %20s %20s %5s %4s %15s %d %d\n", j + 1, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr);

printf("The record has been added.\n");

}

else

{

printf("File not found.\n");

}

}

*// comparator to sort by first name*

**int** asc(**const void \***a, **const void \***b)

{

**struct** student **\***x = (**struct** student **\***)a; **struct** student **\***y = (**struct** student **\***)b; return strcmp(x->fnn, y->fnn);

}

*// display first name*

**void** display\_fname()

{

**int** n, j;

n = no\_of\_rec(); FILE \*fptr;

fptr = fopen("student.txt", "r"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[n]), asc); for (j = 0; j < n; j++)

{

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", stu[j].recno, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr);

}

else

printf("File not found.\n");

}

*// to create index first\_name file*

**void** create\_indexfname()

{

**int** n, j;

n = no\_of\_rec(); FILE \*fptr, \*fptr1;

fptr = fopen("student.txt", "r"); fptr1 = fopen("Index\_Fname.txt", "w"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[0]), asc); for (j = 0; j < n; j++)

{

fprintf(fptr1, "%02d %s\n", stu[j].recno, stu[j].fnn);

}

fclose(fptr); fclose(fptr1);

printf("File Index\_Fname.txt has been created and data has been stored.\n");

}

else

printf("File not found.\n");

}

*// searching student details*

**void** search()

{

FILE \*fptr, \*fptr1;

fptr = fopen("Index\_Fname.txt", "r"); if (fptr != NULL)

{

**char** str[20];

printf("Enter the name you want to search: "); scanf("%s", str);

**int** j = 0, i, found = 0;

**char** temp;

temp = fgetc(fptr);

**int** n = no\_of\_rec(); for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %s\n", &stu[j].recno, stu[j].fnn); if (strcmp(str, stu[j].fnn) == 0)

{

i = stu[j].recno; found = 1;

printf("Record found\n"); break;

}

}

if (found == 1)

{

fptr1 = fopen("student.txt", "r");

fseek(fptr1, (i - 1) \* (sizeof(struct student) + 4), SEEK\_CUR); fscanf(fptr1, "%d %d %s %s %s %s %s %d %d", &stu[0].recno,

&stu[0].idd, stu[0].fnn, stu[0].lnn, stu[0].gn, stu[0].brn, stu[0].city, &stu[0].age, &stu[0].jmarks);

printf("%02d %d %s %s %s %s %s %d %d\n", stu[0].recno, stu[0].idd, stu[0].fnn, stu[0].lnn, stu[0].gn, stu[0].brn, stu[0].city, stu[0].age, stu[0].jmarks);

fclose(fptr); fclose(fptr1);

}

else

printf("Record not found\n");

}

else

printf("File not found\n");

}

*// comparator to sort based on city*

**int** asc1(**const void \***a, **const void \***b)

{

**struct** student **\***x = (**struct** student **\***)a;

**struct** student **\***y = (**struct** student **\***)b; return strcmp(x->city, y->city);

}

**void** city()

{

**int** n, j;

n = no\_of\_rec(); FILE \*fptr;

fptr = fopen("student.txt", "r"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[n]), asc1);

printf("----Using Student file \n");

for (j = 0; j < n; j++)

{

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", stu[j].recno, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr);

}

else

printf("File not found.\n");

}

*// file to create index\_city*

**void** create\_indexcity()

{

**int** n, j;

n = no\_of\_rec(); FILE \*fptr, \*fptr1;

fptr = fopen("student.txt", "r"); fptr1 = fopen("Index\_City.txt", "w"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[0]), asc1);

printf("----Using Index file \n");

for (j = 0; j < n; j++)

{

fprintf(fptr1, "%02d %d %15s %15s %5s %4s %8s %d %d\n", j + 1, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", j + 1, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr); fclose(fptr1);

printf("File Index\_City.txt has been created and data has been stored.\n");

}

else

printf("File not found.\n");

}

**int** asc2(**const void \***a, **const void \***b)

{

**struct** student **\***x = (**struct** student **\***)a; **struct** student **\***y = (**struct** student **\***)b; return (x->age - y->age);

}

**void** city\_age()

{

**int** n, j;

n = no\_of\_rec(); FILE \*fptr;

fptr = fopen("student.txt", "r"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[n]), asc1);

printf("----Without using Index File \n");

printf("\nAccording to City: \n"); for (j = 0; j < n; j++)

{

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", stu[j].recno, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr);

printf("\nAccording to Age: \n");

fptr = fopen("student.txt", "r"); for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[0]), asc2); for (j = 0; j < n; j++)

{

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", stu[j].recno, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr);

}

else

printf("File not found.\n");

}

*// create a file for index\_age*

**void** create\_indexage()

{

**int** n, j;

n = no\_of\_rec(); FILE \*fptr, \*fptr1;

fptr = fopen("student.txt", "r"); fptr1 = fopen("Index\_Age.txt", "w"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

qsort(stu, n, sizeof(stu[0]), asc2);

printf("----Using Index file \n");

for (j = 0; j < n; j++)

{

fprintf(fptr1, "%02d %d %15s %15s %5s %4s %8s %d %d\n", j + 1, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", j + 1, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr); fclose(fptr1);

}

else

printf("File not found.\n");

}

*// using index file displaying data*

**void** index\_cityage()

{

FILE \*fptr, \*fptr1;

fptr = fopen("Index\_City.txt", "r"); if (fptr != NULL)

{

**char** temp;

temp = fgetc(fptr);

**int** j = 0;

printf("----Using Index File \n");

printf("\nAccording to City: \n");

**int** n = no\_of\_rec(); for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d%d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", stu[j].recno, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr); create\_indexage();

fptr = fopen("Index\_Age.txt", "r"); printf("\nAccording to Age: \n"); for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d%d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

printf("%02d %d %15s %15s %5s %4s %8s %d %d\n", stu[j].recno, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr);

}

else

printf("File not found.\n");

}

*// add new record*

**void** add\_update()

{

**int** n, n1, j, i;

n = no\_of\_rec(); FILE \*fptr;

fptr = fopen("student.txt", "r"); if (fptr != NULL)

{

for (j = 0; j < n; j++)

{

fscanf(fptr, "%d %d %s %s %s %s %s %d %d", &stu[j].recno, &stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, &stu[j].age, &stu[j].jmarks);

}

fclose(fptr);

printf("File found succesfully.\n"); printf("Enter the no. of records to be added: "); scanf("%d", &n1);

for (i = 0; i < n1; i++)

{

printf("Enter id: "); scanf("%d", &stu[n + i].idd); printf("Enter first name: "); scanf("%s", stu[n + i].fnn); printf("Enter last name: "); scanf("%s", stu[n + i].lnn); printf("Enter gender: "); scanf("%s", stu[n + i].gn); printf("Enter branch: "); scanf("%s", stu[n + i].brn); printf("Enter city: "); scanf("%s", stu[n + i].city); printf("Enter age: "); scanf("%d", &stu[n + i].age); printf("Enter jee marks: "); scanf("%d", &stu[n + i].jmarks);

}

qsort(stu, n + n1, sizeof(stu[0]), dsc); fptr = fopen("student.txt", "w");

for (j = 0; j < n + n1; j++)

{

fprintf(fptr, "%02d %d %20s %20s %5s %4s %15s %d %d\n", j + 1, stu[j].idd, stu[j].fnn, stu[j].lnn, stu[j].gn, stu[j].brn, stu[j].city, stu[j].age, stu[j].jmarks);

}

fclose(fptr);

printf("The records have been added.\n"); create\_indexfname();

}

else

{

printf("File not found.\n");

}

}

**int** main()

{

**clock\_t** t;

**double** time\_taken;

**int** ch;

**char** choice; do

{

*// lists*

printf("\n");

printf("1.Create.\n"); printf("2.Display records.\n");

printf("3.Add a record in the sequential file of students.\n"); printf("4.Display list of students as per their ascending order of

FNAME.\n");

printf("5.Create fname index file and store data\n"); printf("6.Search for the student's FNAME and diaplay.\n"); printf("7.Execution time to display records city wise.\n"); printf("8.Display all the student records ascending order of city and

age.\n");

printf("9.Add and Update Index File.\n"); printf("0.Exit\n");

printf("Enter your choice: "); scanf("%d", &ch);

**int** n, k; switch (ch)

{

case 1:

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

create(n); break;

case 2:

display(); break;

case 3:

add(); break;

case 4:

display\_fname(); break;

case 5:

create\_indexfname(); break;

case 6:

create\_indexfname(); search();

break; case 7:

do

{

*//comparision*

printf("\na.Execution time involving Student file.\n"); printf("b.Execution time involving Index City file.\n"); printf("c.Exit\n");

printf("Enter your choice: "); fflush(stdin);

scanf("%c", &choice); switch (choice)

{

case 'a':

t = clock(); city();

t = clock() - t;

time\_taken = ((**double**)t) / CLOCKS\_PER\_SEC; printf("Execution Time:%f secs \n", time\_taken); break;

case 'b':

t = clock(); create\_indexcity(); t = clock() - t;

time\_taken = ((**double**)t) / CLOCKS\_PER\_SEC; printf("Execution Time:%f secs \n", time\_taken); break;

default:

break;

}

} while (choice != 'c'); break;

case 8:

do

{

printf("\na.Display without involving index file.\n"); printf("b.Display involving Index City file.\n"); printf("c.Exit\n");

printf("Enter your choice: "); fflush(stdin);

scanf("%c", &choice); switch (choice)

{

case 'a':

city\_age();

break;

case 'b':

create\_indexcity(); index\_cityage(); break;

default:

break;

}

} while (choice != 'c'); break;

case 9:

add\_update(); break;

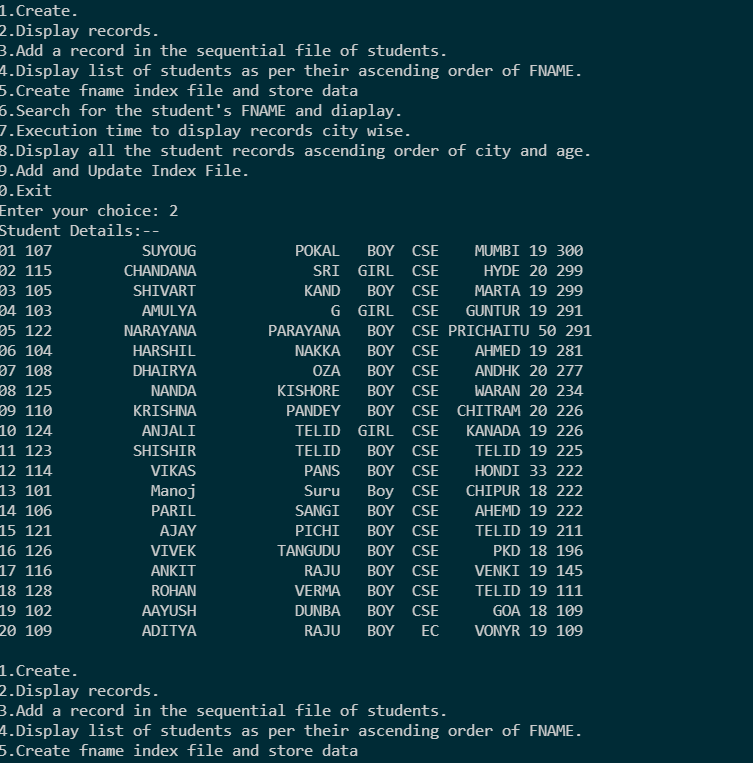
default:

break;

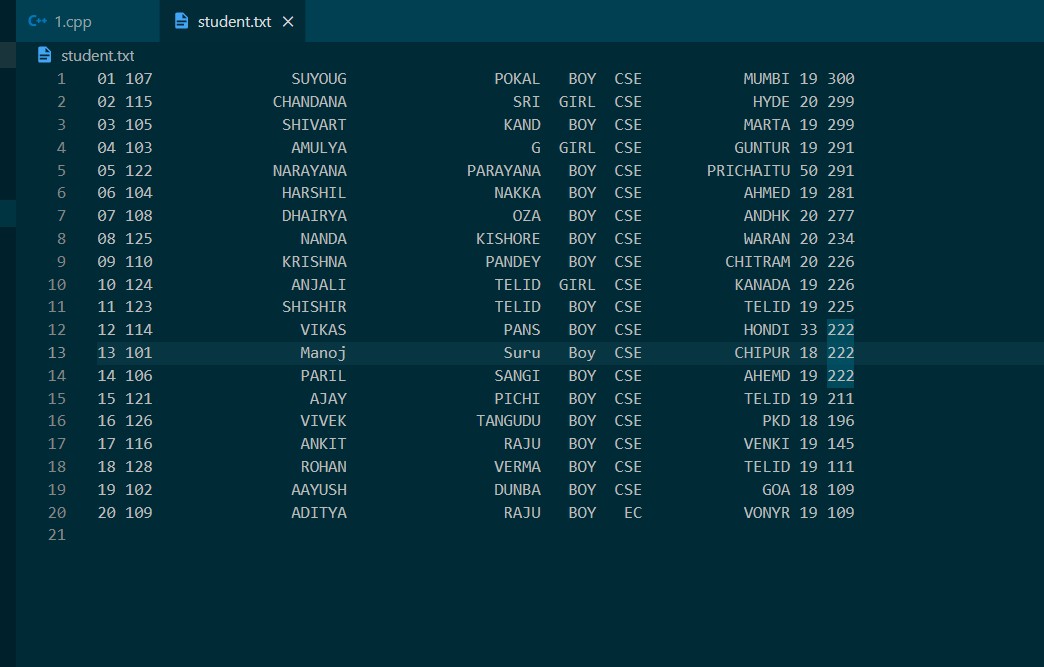
}

} while (ch != 0); return 0;

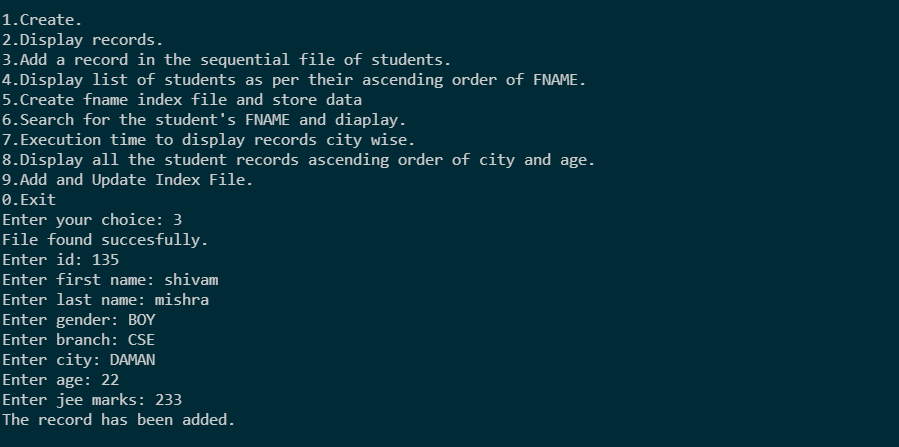
}



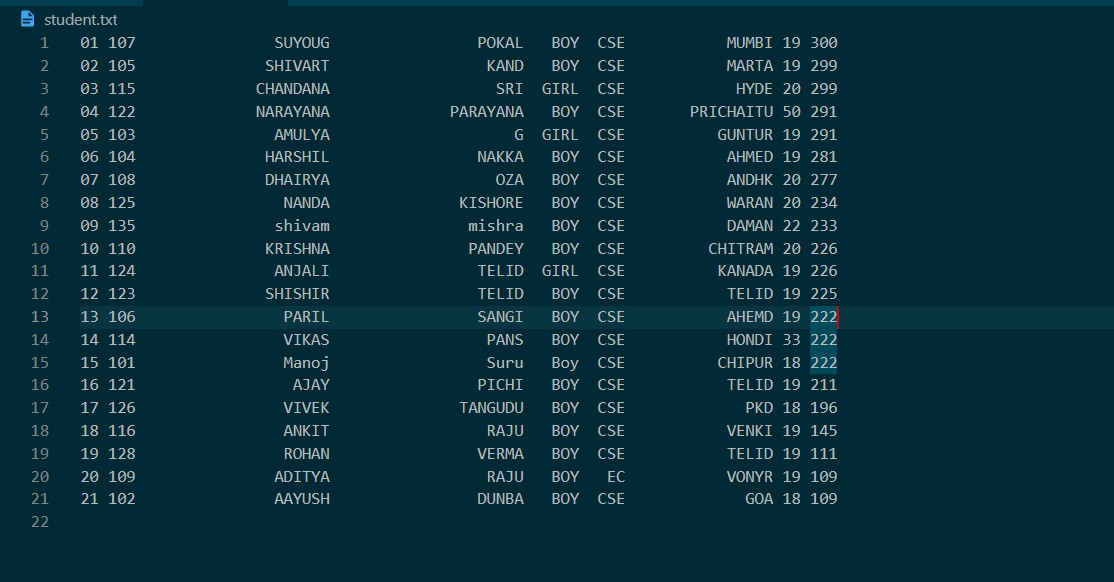
Student.txt



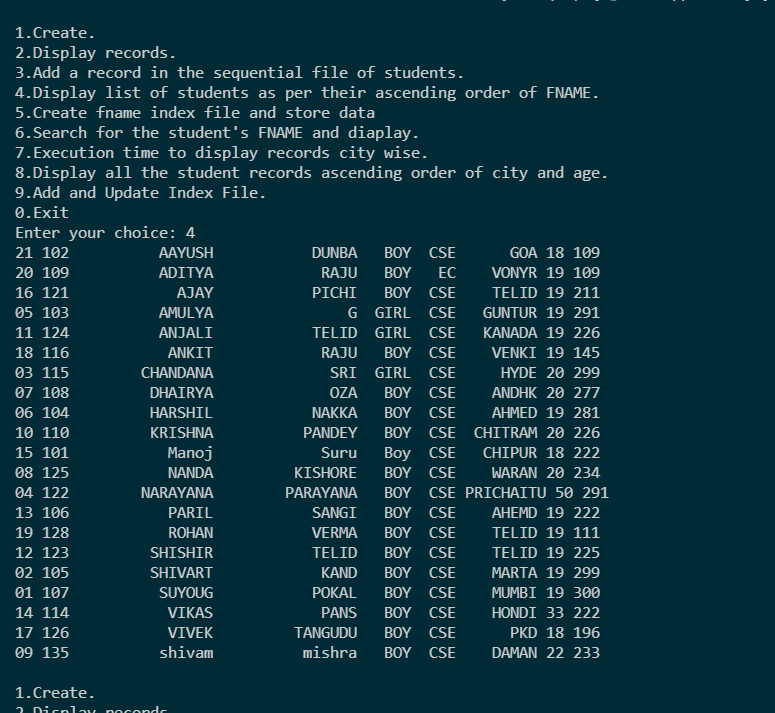
1. Add a record in the sequential file of students.



STUDENT.TXT



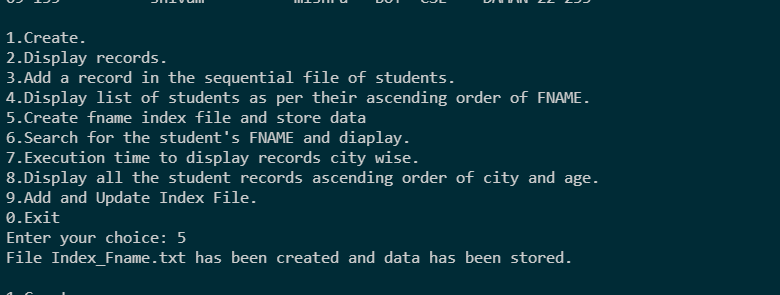
1. Display list of students as per their ascending order of FNAME.



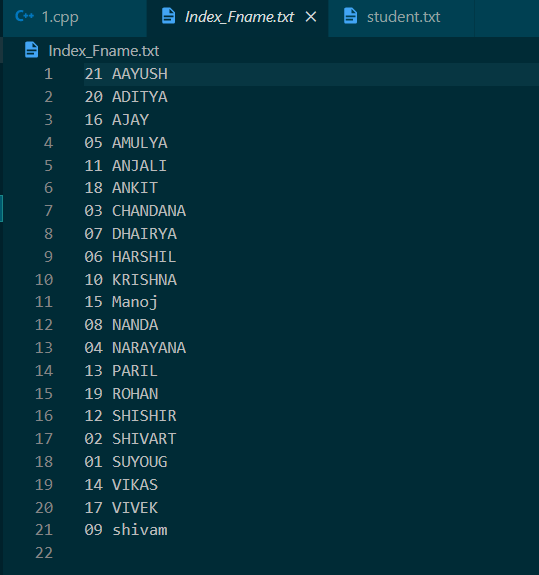
Displaying list of students as per their ascending order of FNAME.

1. Create an index file on FNAME with file name: “Index\_Fname.txt” consists of sorted Fname

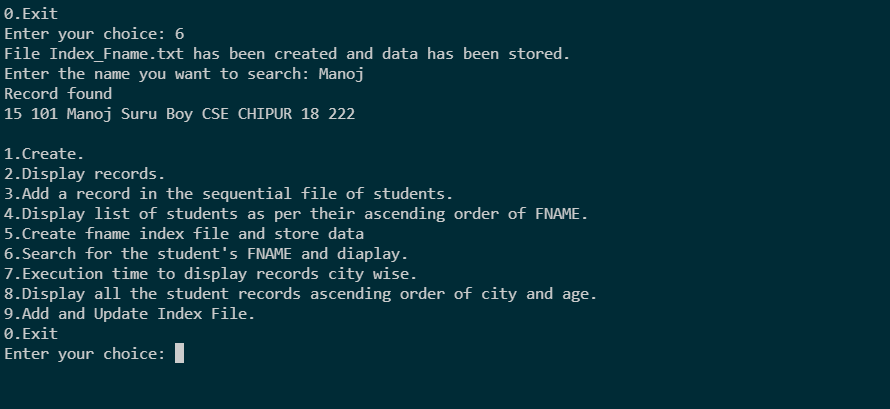
and the Record number in the student file.



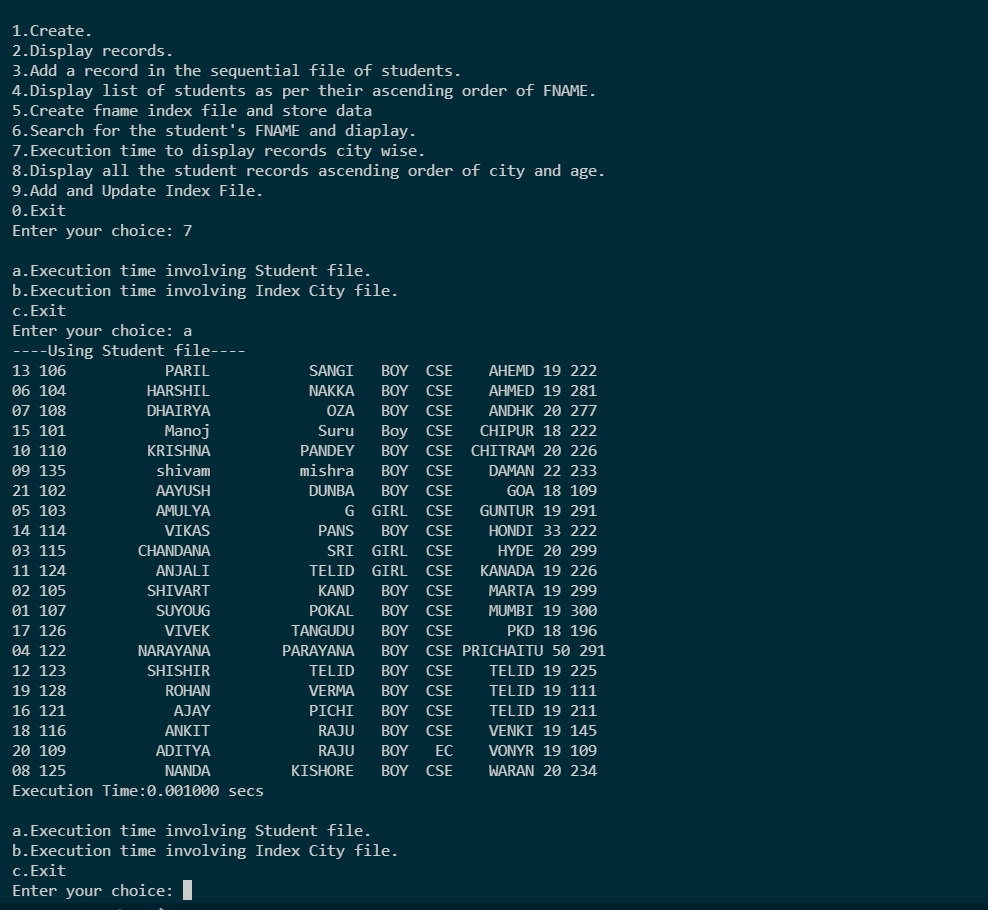
Index\_Fname.txt



1. Search for the student’s FNAME using INDEX FILE (“Index\_Fname.txt”) and display the record from the student file directly by setting the file pointer to the position= record number \* size of student structure.

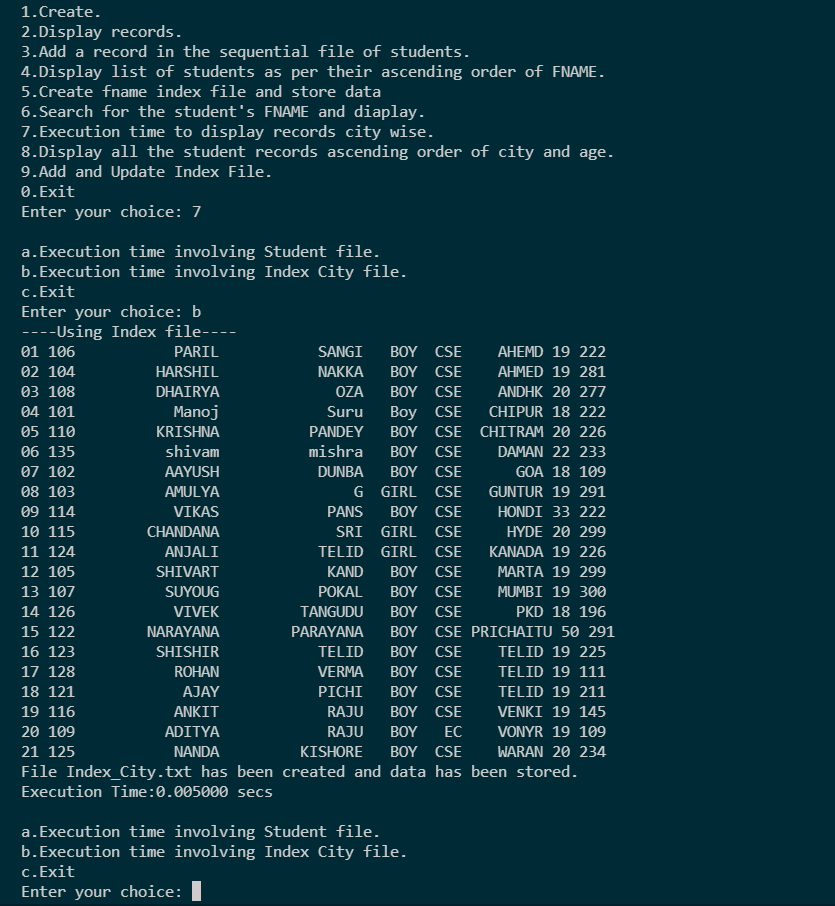


1. a. Show the execution time to display city wise the records of students from the sequential file of student.

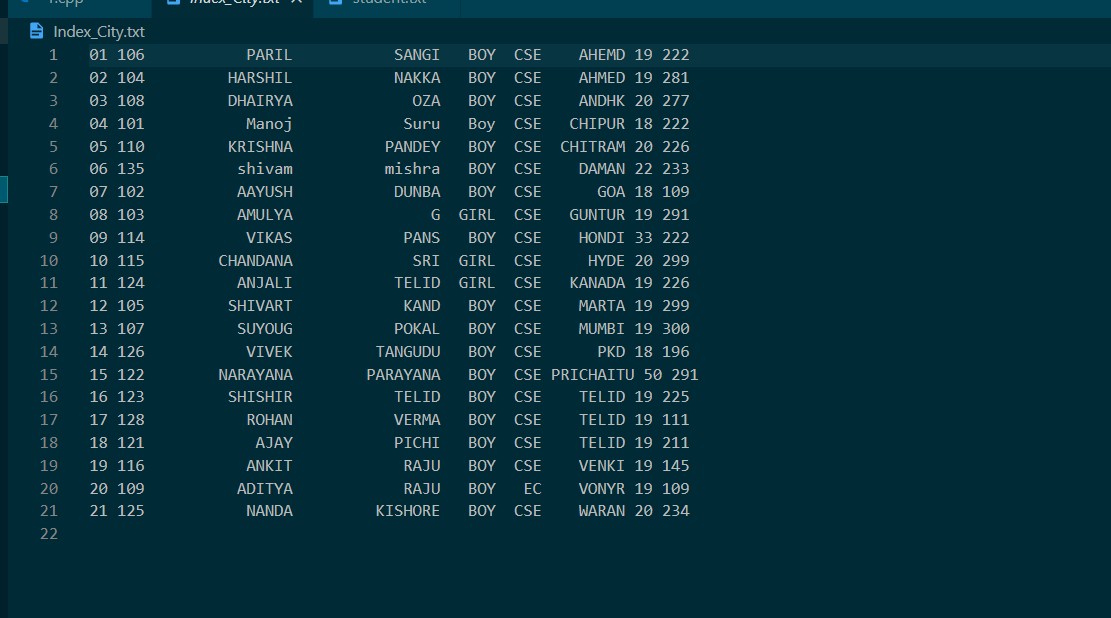


b. Show the execution time to display city wise the records of students by creating the

student’s City using INDEX FILE (“Index\_City.txt”).

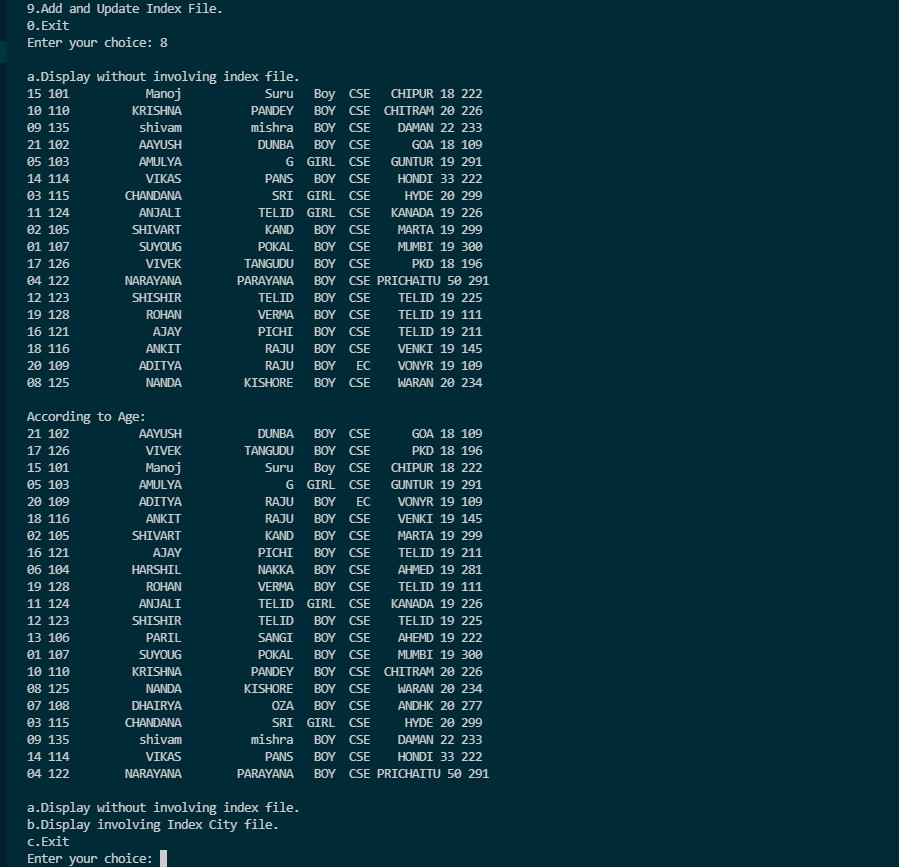


Index\_City.txt

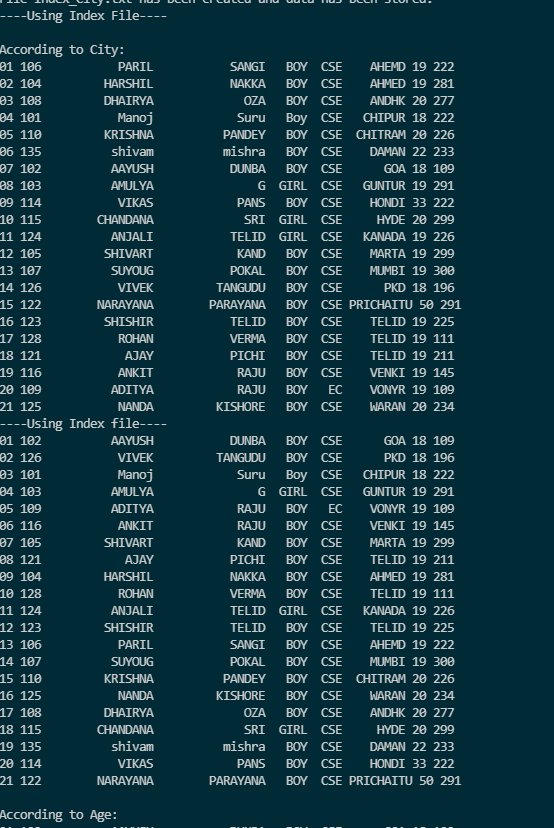


1. List all the student records of the file in ascending order of city and age using index and without index file of CITY and AGE.

Without using index file

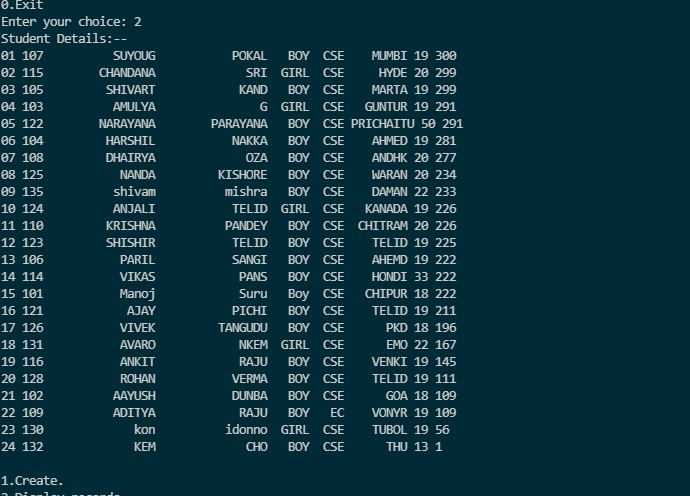


Using Index File:



1. Insert some more records in student file and update the index file of fname to manage the ordered information.

AFTER ADDING 3 RECORDS



INDEX\_FNAME.TXT

