Name : Aritra Das

Roll No: 75

Assignment 2

1.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <stdbool.h>

struct node

{

    int roll;

    char name[30];

    int score;

};

void screen()

{

    printf("\n");

    printf("1 search for a data in the file.\n");

    printf("2 add data to the file.\n");

    printf("3 edit any data in the file.\n");

    printf("4 delete any data in the file.\n");

    printf("5 display the contents of the file.\n");

    printf("6 exit.\n");

}

struct node search (int type, bool \*ok, int key)

{

    FILE \*fin;

    fin = fopen("info.dat", "rb");

    struct node tp;

    if (fin == NULL) {

        printf("Error occured while opening file...\n");

        return tp;

    }

    while (fread(&tp, sizeof(struct node), 1, fin)) {

        if (tp.roll == key) {

            \*ok = true;

            break;

        }

    }

    if (type == 1) {

        if (\*ok) {

            printf("Data with roll : %d is found ", key);

            printf("[%d, \'%s\', %d]\n", tp.roll, tp.name, tp.score);

        } else {

            printf("Data with roll : %d couldn't be found.\n", key);

        }

    }

    return tp;

}

void add ()

{

    FILE \*fout;

    fout = fopen("info.dat", "ab");

    if (fout == NULL) {

        printf("Error occured while opening file...\n");

        return;

    }

    struct node newNode;

    printf("Enter data to be added : \n");

    printf("Enter roll : \n");

    scanf("%d", &newNode.roll);

    bool ok = false;

    search(0, &ok, newNode.roll);

    if (ok) {

        printf("This roll number already exists in the file.\n");

        printf("Please enter a distinct roll number.\n");

        return;

    } else {

        printf("Enter name : \n");

        scanf(" %[^\n]s", newNode.name);

        printf("Enter score : \n");

        scanf("%d", &newNode.score);

        fwrite(&newNode, sizeof(struct node), 1, fout);

        if (fwrite != 0) {

            printf("Data added successfully.\n");

        } else {

            printf("Error writing file. Data couldn\'t be added.\n");

        }

    }

    fclose(fout);

}

void edit ()

{

    printf("Enter roll to edit : \n");

    int key;

    scanf("%d", &key);

    bool ok = false;

    struct node change = search(1, &ok, key);

    if (ok) {

        char choice = 'n';

        printf("Do you want to edit name? (y/n)\n");

        scanf(" %c", &choice);

        if (choice == 'y') {

            printf("Enter new name : \n");

            scanf(" %[^\n]s", change.name);

        }

        choice = 'n';

        printf("Do you want to edit score? (y/n)\n");

        scanf(" %c", &choice);

        if (choice == 'y') {

            printf ("Enter new score : \n");

            scanf("%d", &change.score);

        }

        FILE \*fin, \*tp;

        tp = fopen("tp.dat", "wb");

        fin = fopen("info.dat", "rb");

        if (tp == NULL || fin == NULL) {

            printf("Error occured while opening file...\n");

        }

        struct node curr;

        while (fread(&curr, sizeof(struct node), 1, fin)) {

            fwrite(&curr, sizeof(struct node), 1, tp);

        }

        fclose(tp);

        fclose(fin);

        FILE \*fout;

        tp = fopen("tp.dat", "rb");

        fout = fopen("info.dat", "wb");

        if (fout == NULL) {

            printf("Error occured while opening file...\n");

        }

        while (fread(&curr, sizeof(struct node), 1, tp)) {

            if (curr.roll == key) {

                fwrite(&change, sizeof(struct node), 1, fout);

            } else {

                fwrite(&curr, sizeof(struct node), 1, fout);

            }

        }

        fclose(tp);

        fclose(fout);

        printf("File Edited .\n");

    }

}

void delete ()

{

    printf("Enter roll to delete : \n");

    int key;

    scanf("%d", &key);

    bool ok = false;

    struct node change = search(1, &ok, key);

    if (ok) {

        FILE \*fin, \*tp;

        tp = fopen("tp.dat", "wb");

        fin = fopen("info.dat", "rb");

        if (tp == NULL || fin == NULL) {

            printf("Error occured while opening file...\n");

        }

        struct node curr;

        while (fread(&curr, sizeof(struct node), 1, fin)) {

            fwrite(&curr, sizeof(struct node), 1, tp);

        }

        fclose(tp);

        fclose(fin);

        FILE \*fout;

        tp = fopen("tp.dat", "rb");

        fout = fopen("info.dat", "wb");

        if (fout == NULL) {

            printf("Error occured while opening file...\n");

        }

        while (fread(&curr, sizeof(struct node), 1, tp)) {

            if (curr.roll == key) {

                continue;

            } else {

                fwrite(&curr, sizeof(struct node), 1, fout);

            }

        }

        fclose(tp);

        fclose(fout);

        printf("Data Deleted Successfully. File Edited Successfully.\n");

    }

}

void display ()

{

    FILE \*fin;

    fin = fopen("info.dat", "rb");

    if (fin == NULL) {

        printf("Error occured while opening file");

        return;

    }

    struct node curr;

    printf("File contents are : \n");

    while (fread(&curr, sizeof(struct node), 1, fin)) {

        printf("[%d, \'%s\', %d]\n", curr.roll, curr.name, curr.score);

    }

}

int main()

{

    int s;

    do {

        screen();

        scanf("%d", &s);

        bool ok = false;

        int key;

        switch (s) {

            case 1 :    printf("Enter roll number to be searched : \n");

                        scanf("%d", &key);

                        search(1, &ok, key);

                        break;

            case 2 :    add();

                        break;

            case 3 :    edit();

                        break;

            case 4 :    delete();

                        break;

            case 5 :    display();

                        break;

            case 6 :

                        break;

            default :   printf("Invalid Choice. Please enter a valid choice.\n");

                        break;

        }

    } while (s != 6);

    return 0;

}

