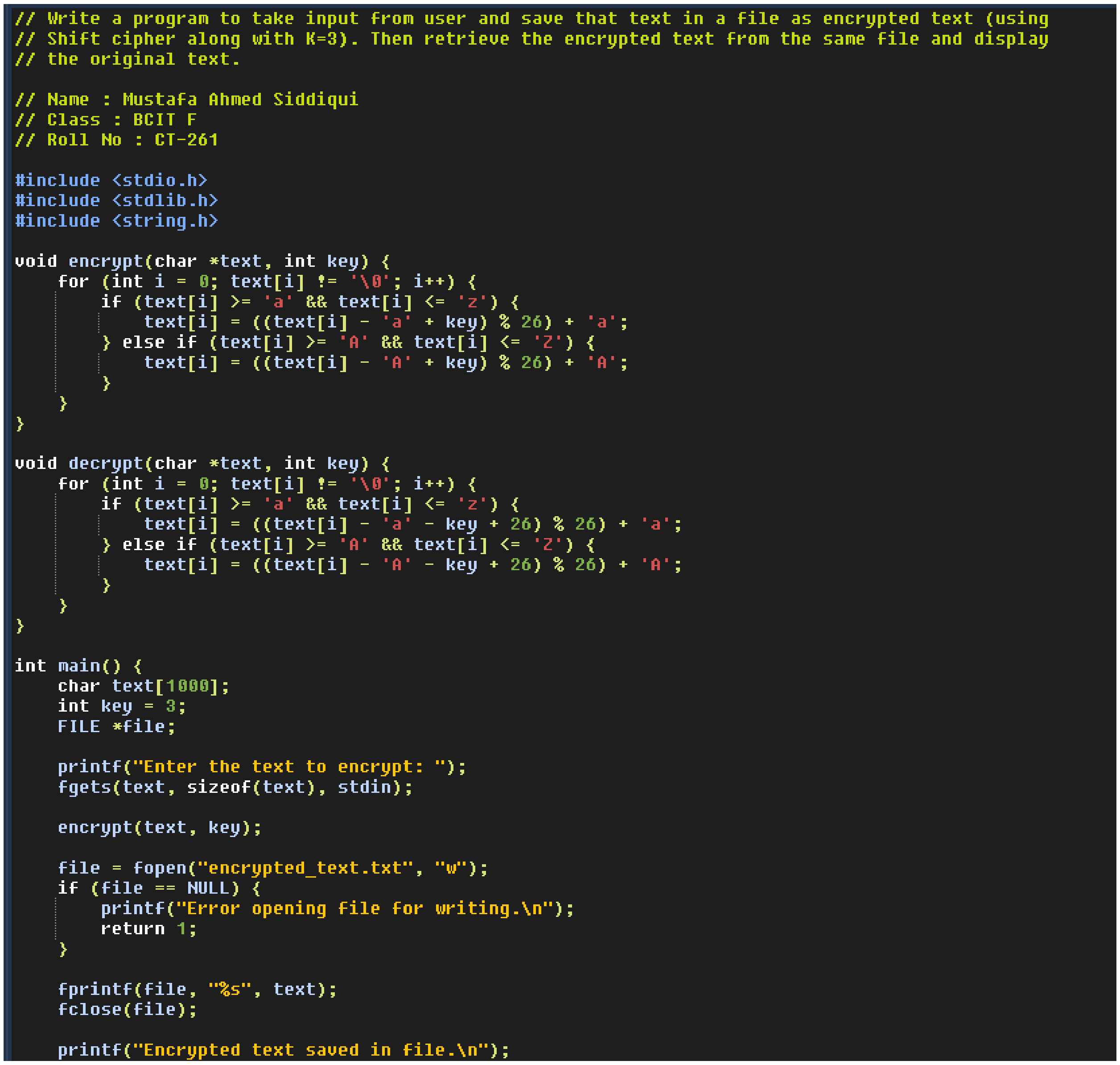
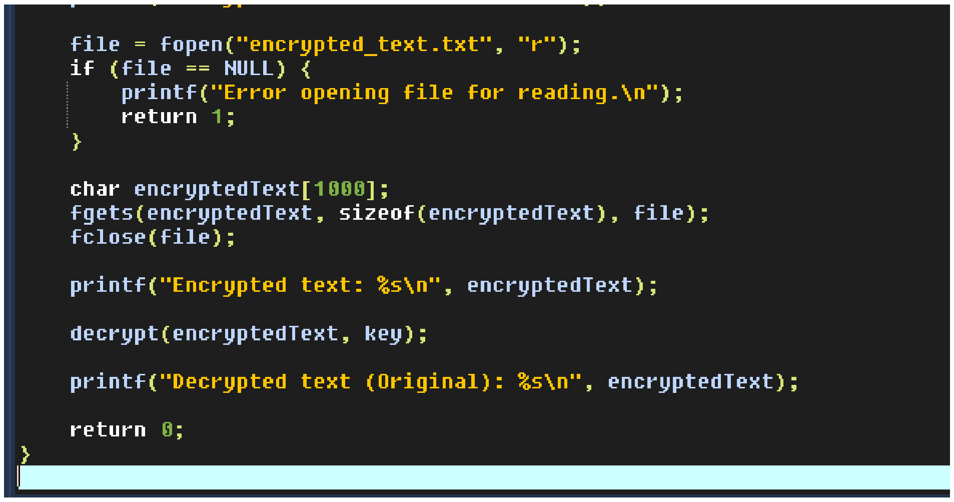
PF LAB TASK 13

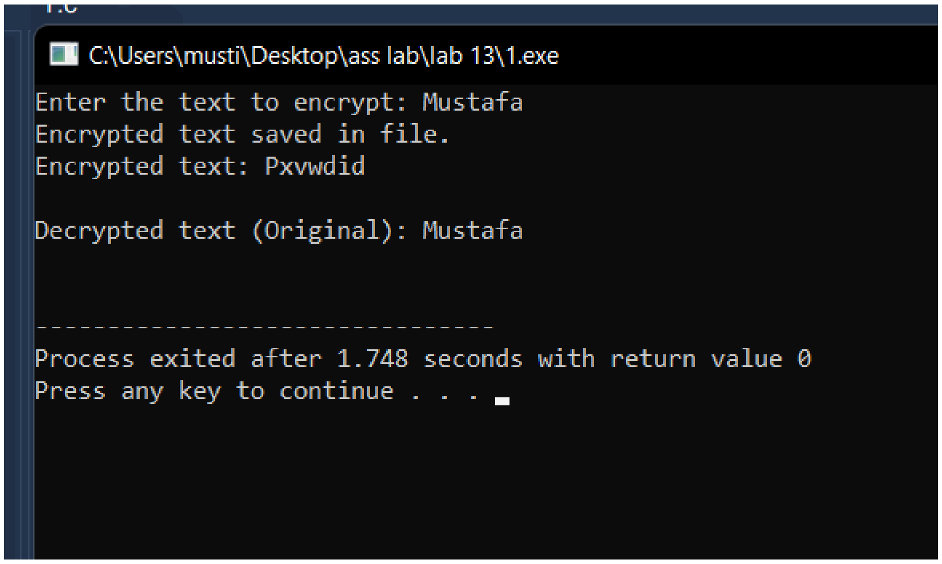
1. Write a program to take input from user and save that text in a file as encrypted text (using Shift cipher along with K=3). Then retrieve the encrypted text from the same file and display the original text.



```c
// Write a program to take input from user and save that text in a file as encrypted text (using
// Shift cipher along with K=3). Then retrieve the encrypted text from the same file and display
// the original text.

// Name : Mustafa Ahmed Siddiqui
// Class : BCIT F
// Roll No : CT-261

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

void encrypt(char *text, int key) {
    for (int i = 0; text[i] != '\0'; i++) {
        if (text[i] >= 'a' && text[i] <= 'z') {
            text[i] = ((text[i] - 'a' + key) % 26) + 'a';
        } else if (text[i] >= 'A' && text[i] <= 'Z') {
            text[i] = ((text[i] - 'A' + key) % 26) + 'A';
        }
    }
}

void decrypt(char *text, int key) {
    for (int i = 0; text[i] != '\0'; i++) {
        if (text[i] >= 'a' && text[i] <= 'z') {
            text[i] = ((text[i] - 'a' - key + 26) % 26) + 'a';
        } else if (text[i] >= 'A' && text[i] <= 'Z') {
            text[i] = ((text[i] - 'A' - key + 26) % 26) + 'A';
        }
    }
}

int main() {
    char text[1000];
    int key = 3;
    FILE *file;

    printf("Enter the text to encrypt: ");
    fgets(text, sizeof(text), stdin);

    encrypt(text, key);

    file = fopen("encrypted_text.txt", "w");
    if (file == NULL) {
        printf("Error opening file for writing.\n");
        return 1;
    }

    fprintf(file, "%s", text);
    fclose(file);

    printf("Encrypted text saved in file.\n");
```

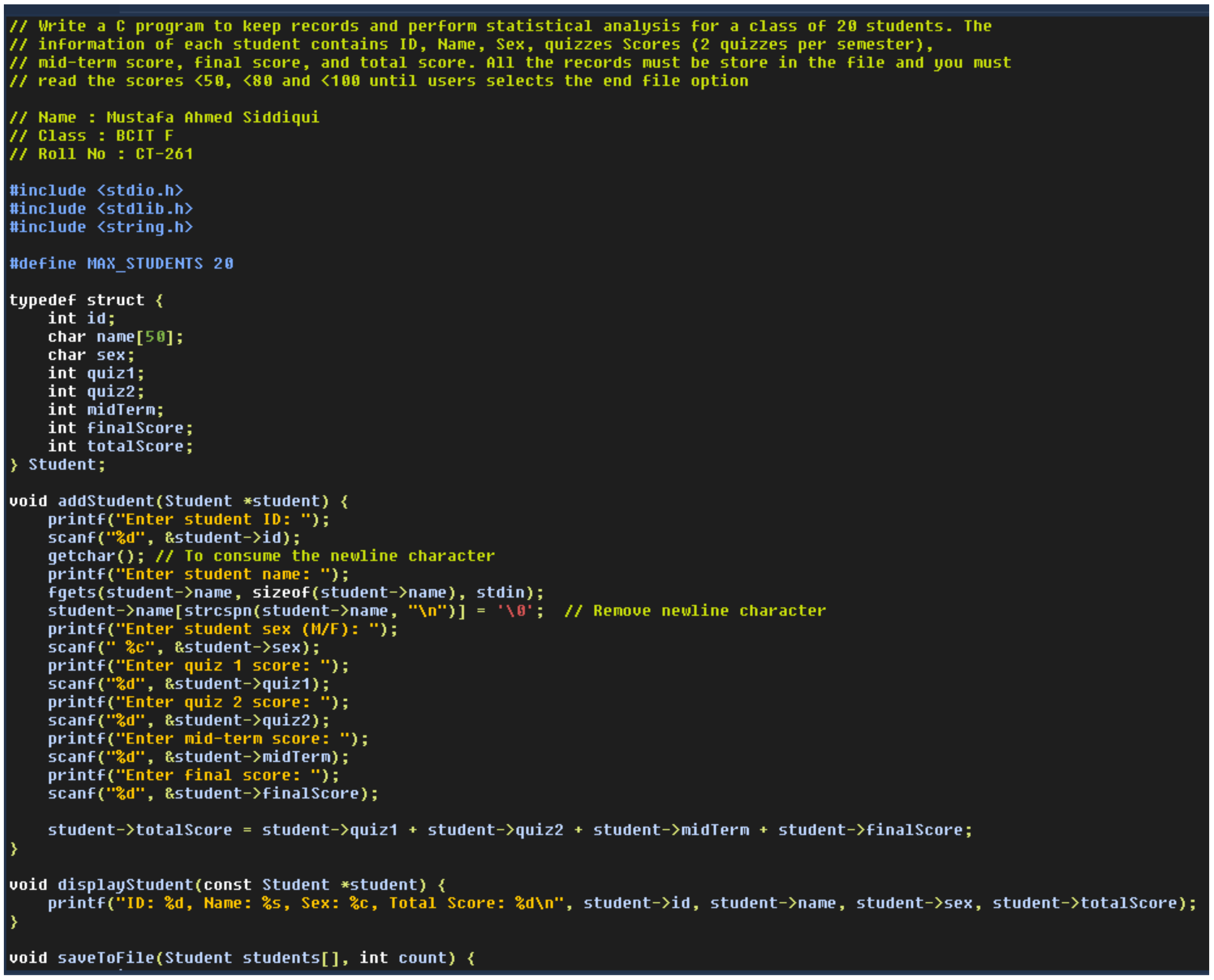


```c
    file = fopen("encrypted_text.txt", "r");
    if (file == NULL) {
        printf("Error opening file for reading.\n");
        return 1;
    }

    char encryptedText[1000];
    fgets(encryptedText, sizeof(encryptedText), file);
    fclose(file);

    printf("Encrypted text: %s\n", encryptedText);

    decrypt(encryptedText, key);

    printf("Decrypted text (Original): %s\n", encryptedText);

    return 0;
}
```

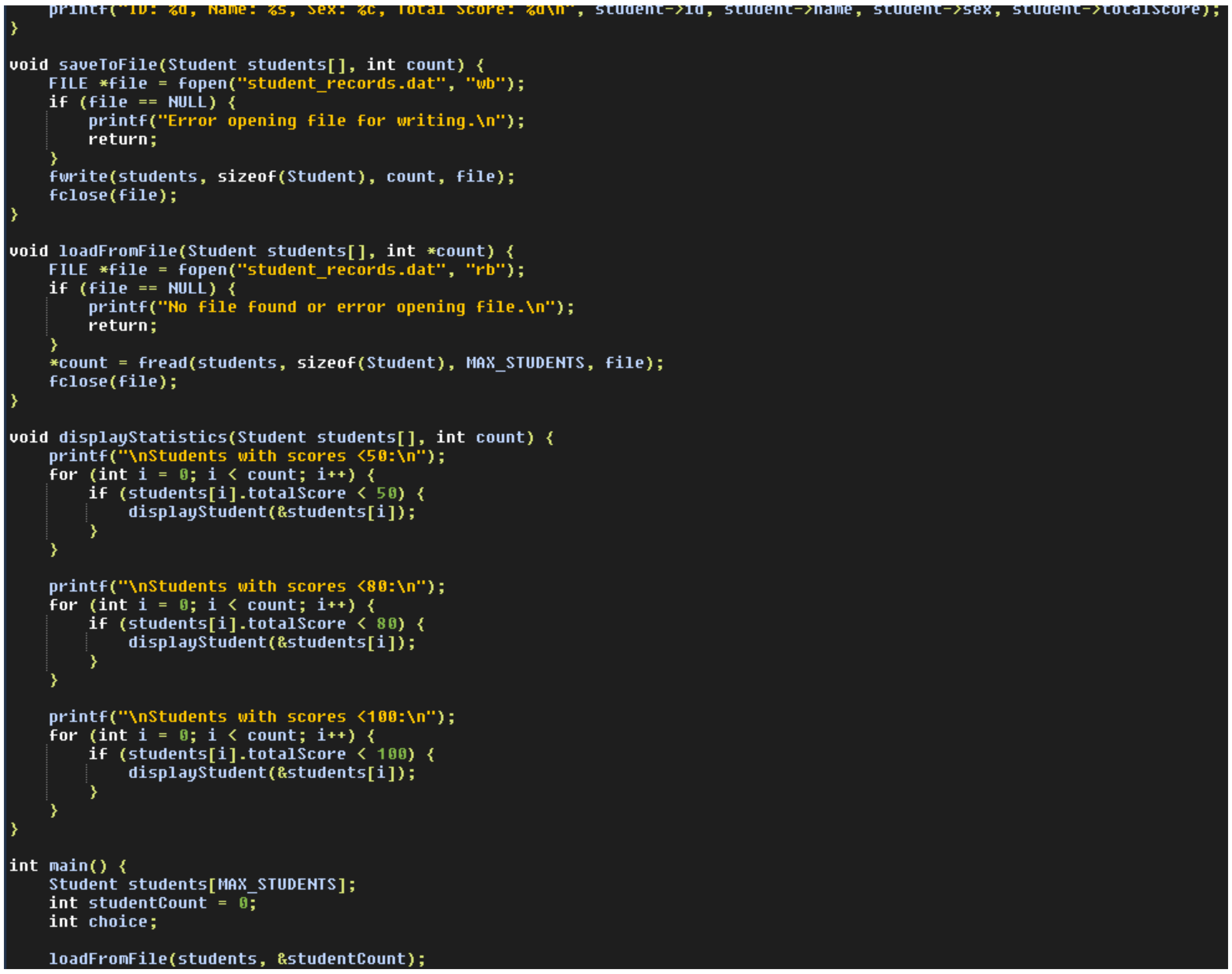


```
C:\Users\musti\Desktop\ass lab\lab 13\1.exe
Enter the text to encrypt: Mustafa
Encrypted text saved in file.
Encrypted text: Pxvwdid

Decrypted text (Original): Mustafa



--------------------------------
Process exited after 1.748 seconds with return value 0
Press any key to continue . . .
```

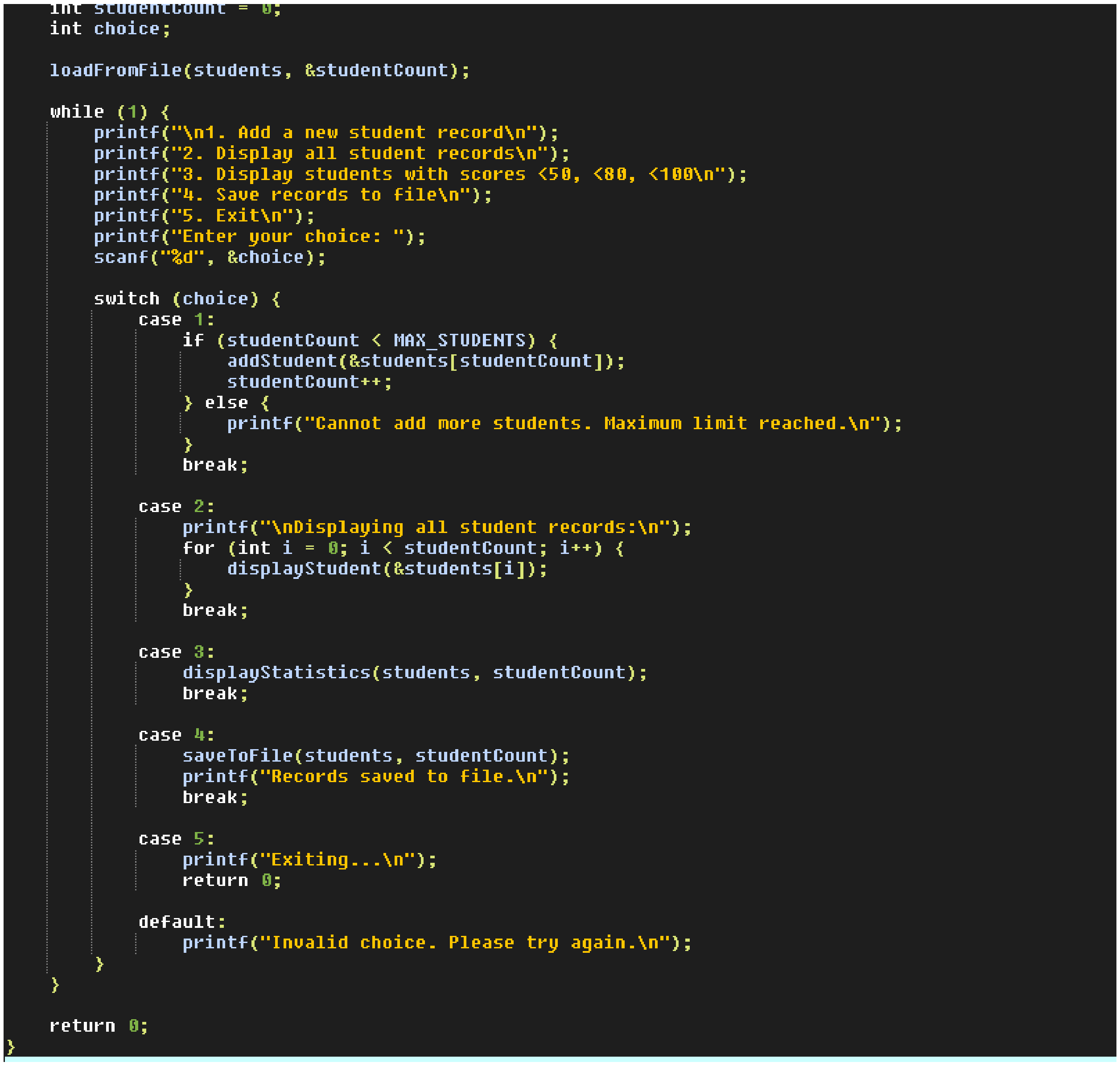
2. Write a C program to keep records and perform statistical analysis for a class of 20 students. The information of each student contains ID, Name, Sex, quizzes Scores (2 quizzes per semester), mid-term score, final score, and total score. All the records must be store in the file and you must read the scores



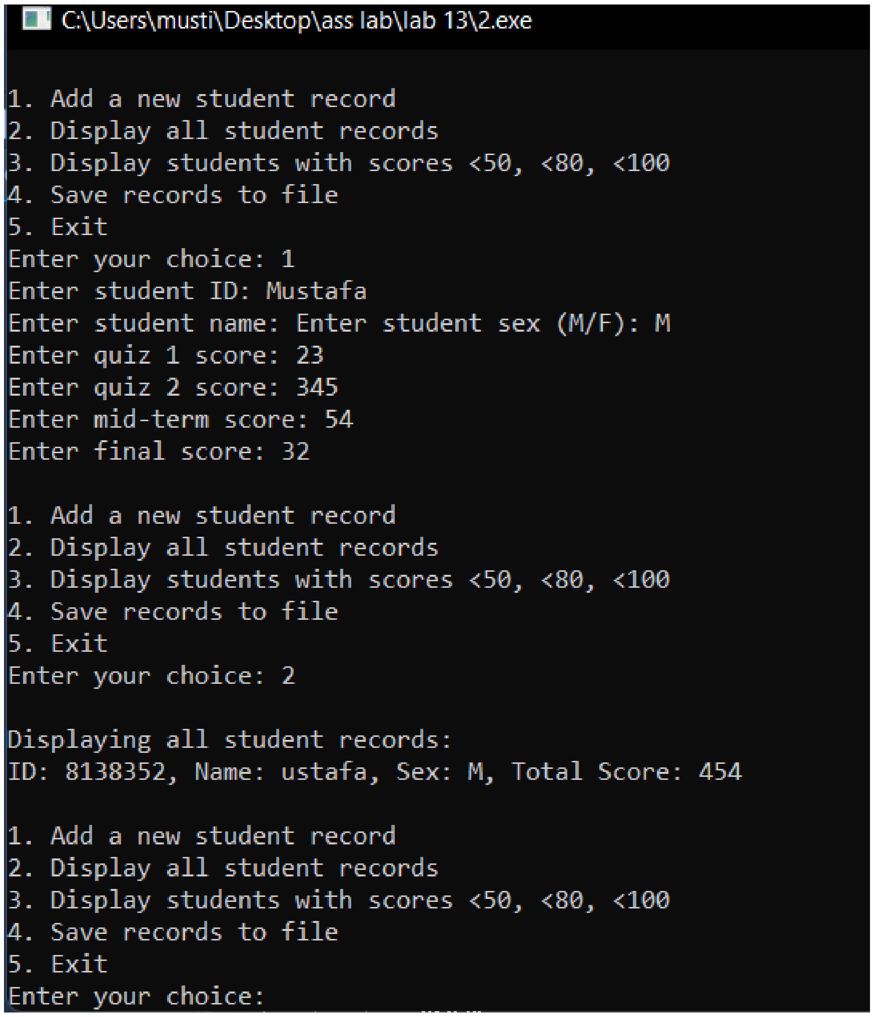
```c
// Write a C program to keep records and perform statistical analysis for a class of 20 students. The
// information of each student contains ID, Name, Sex, quizzes Scores (2 quizzes per semester),
// mid-term score, final score, and total score. All the records must be store in the file and you must
// read the scores <50, <80 and <100 until users selects the end file option

// Name : Mustafa Ahmed Siddiqui
// Class : BCIT F
// Roll No : CT-261

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

#define MAX_STUDENTS 20

typedef struct {
    int id;
    char name[50];
    char sex;
    int quiz1;
    int quiz2;
    int midTerm;
    int finalScore;
    int totalScore;
} Student;

void addStudent(Student *student) {
    printf("Enter student ID: ");
    scanf("%d", &student->id);
    getchar(); // To consume the newline character
    printf("Enter student name: ");
    fgets(student->name, sizeof(student->name), stdin);
    student->name[strcspn(student->name, "\n")] = '\0';  // Remove newline character
    printf("Enter student sex (M/F): ");
    scanf(" %c", &student->sex);
    printf("Enter quiz 1 score: ");
    scanf("%d", &student->quiz1);
    printf("Enter quiz 2 score: ");
    scanf("%d", &student->quiz2);
    printf("Enter mid-term score: ");
    scanf("%d", &student->midTerm);
    printf("Enter final score: ");
    scanf("%d", &student->finalScore);

    student->totalScore = student->quiz1 + student->quiz2 + student->midTerm + student->finalScore;
}

void displayStudent(const Student *student) {
    printf("ID: %d, Name: %s, Sex: %c, Total Score: %d\n", student->id, student->name, student->sex, student->totalScore);
}

void saveToFile(Student students[], int count) {
```



```c
    printf("ID: %d, Name: %s, Sex: %c, Total Score: %d\n", student->id, student->name, student->sex, student->totalScore);
}

void saveToFile(Student students[], int count) {
    FILE *file = fopen("student_records.dat", "wb");
    if (file == NULL) {
        printf("Error opening file for writing.\n");
        return;
    }
    fwrite(students, sizeof(Student), count, file);
    fclose(file);
}

void loadFromFile(Student students[], int *count) {
    FILE *file = fopen("student_records.dat", "rb");
    if (file == NULL) {
        printf("No file found or error opening file.\n");
        return;
    }
    *count = fread(students, sizeof(Student), MAX_STUDENTS, file);
    fclose(file);
}

void displayStatistics(Student students[], int count) {
    printf("\nStudents with scores <50:\n");
    for (int i = 0; i < count; i++) {
        if (students[i].totalScore < 50) {
            displayStudent(&students[i]);
        }
    }

    printf("\nStudents with scores <80:\n");
    for (int i = 0; i < count; i++) {
        if (students[i].totalScore < 80) {
            displayStudent(&students[i]);
        }
    }

    printf("\nStudents with scores <100:\n");
    for (int i = 0; i < count; i++) {
        if (students[i].totalScore < 100) {
            displayStudent(&students[i]);
        }
    }
}

int main() {
    Student students[MAX_STUDENTS];
    int studentCount = 0;
    int choice;

    loadFromFile(students, &studentCount);
```

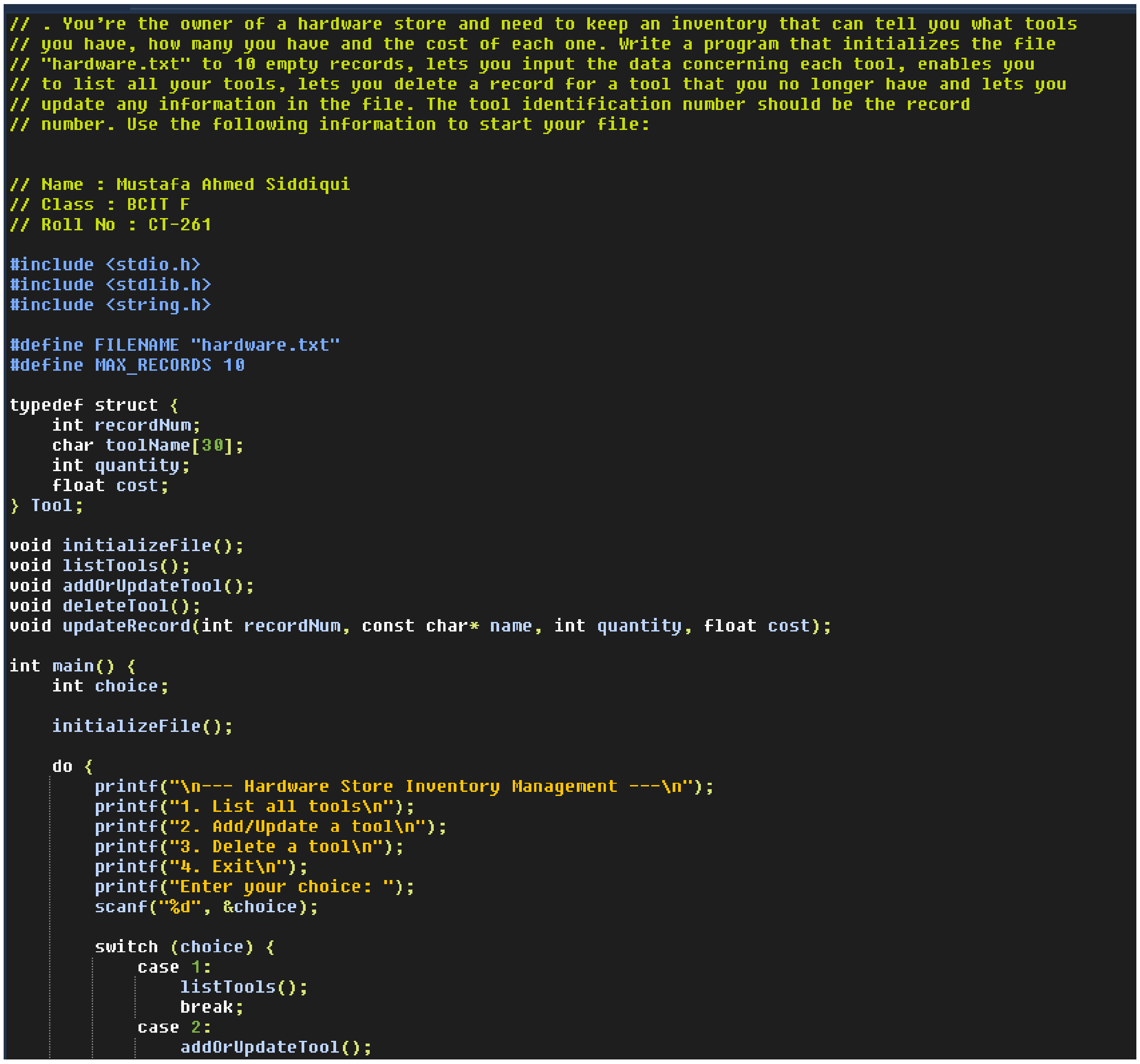


```c
    int studentCount = 0;
    int choice;

    loadFromFile(students, &studentCount);

    while (1) {
        printf("\n1. Add a new student record\n");
        printf("2. Display all student records\n");
        printf("3. Display students with scores <50, <80, <100\n");
        printf("4. Save records to file\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                if (studentCount < MAX_STUDENTS) {
                    addStudent(&students[studentCount]);
                    studentCount++;
                } else {
                    printf("Cannot add more students. Maximum limit reached.\n");
                }
                break;

            case 2:
                printf("\nDisplaying all student records:\n");
                for (int i = 0; i < studentCount; i++) {
                    displayStudent(&students[i]);
                }
                break;

            case 3:
                displayStatistics(students, studentCount);
                break;

            case 4:
                saveToFile(students, studentCount);
                printf("Records saved to file.\n");
                break;

            case 5:
                printf("Exiting...\n");
                return 0;

            default:
                printf("Invalid choice. Please try again.\n");
        }
    }

    return 0;
}
```

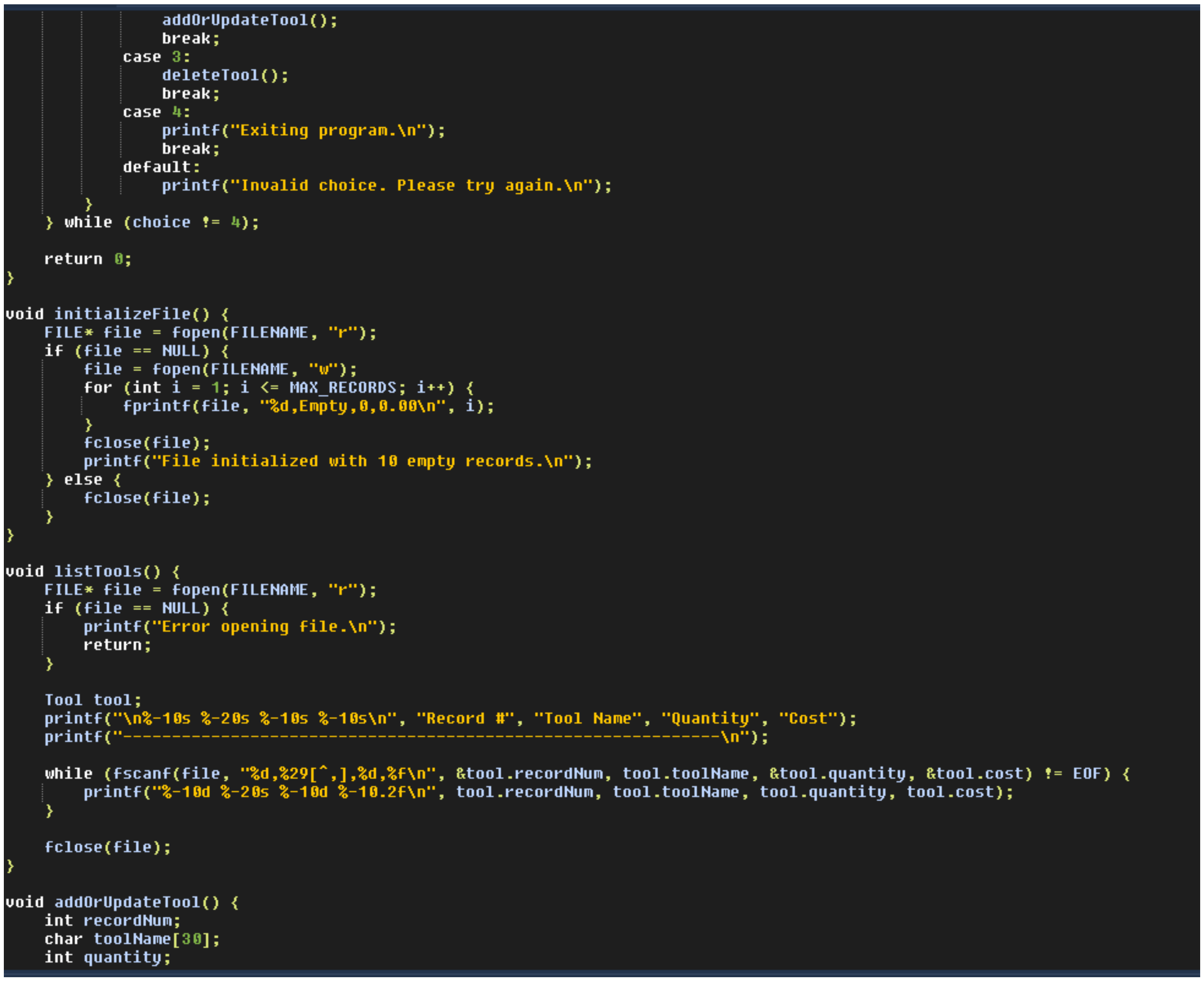


```
C:\Users\musti\Desktop\ass lab\lab 13\2.exe

1. Add a new student record
2. Display all student records
3. Display students with scores <50, <80, <100
4. Save records to file
5. Exit
Enter your choice: 1
Enter student ID: Mustafa
Enter student name: Enter student sex (M/F): M
Enter quiz 1 score: 23
Enter quiz 2 score: 345
Enter mid-term score: 54
Enter final score: 32

1. Add a new student record
2. Display all student records
3. Display students with scores <50, <80, <100
4. Save records to file
5. Exit
Enter your choice: 2

Displaying all student records:
ID: 8138352, Name: ustafa, Sex: M, Total Score: 454

1. Add a new student record
2. Display all student records
3. Display students with scores <50, <80, <100
4. Save records to file
5. Exit
Enter your choice:
```

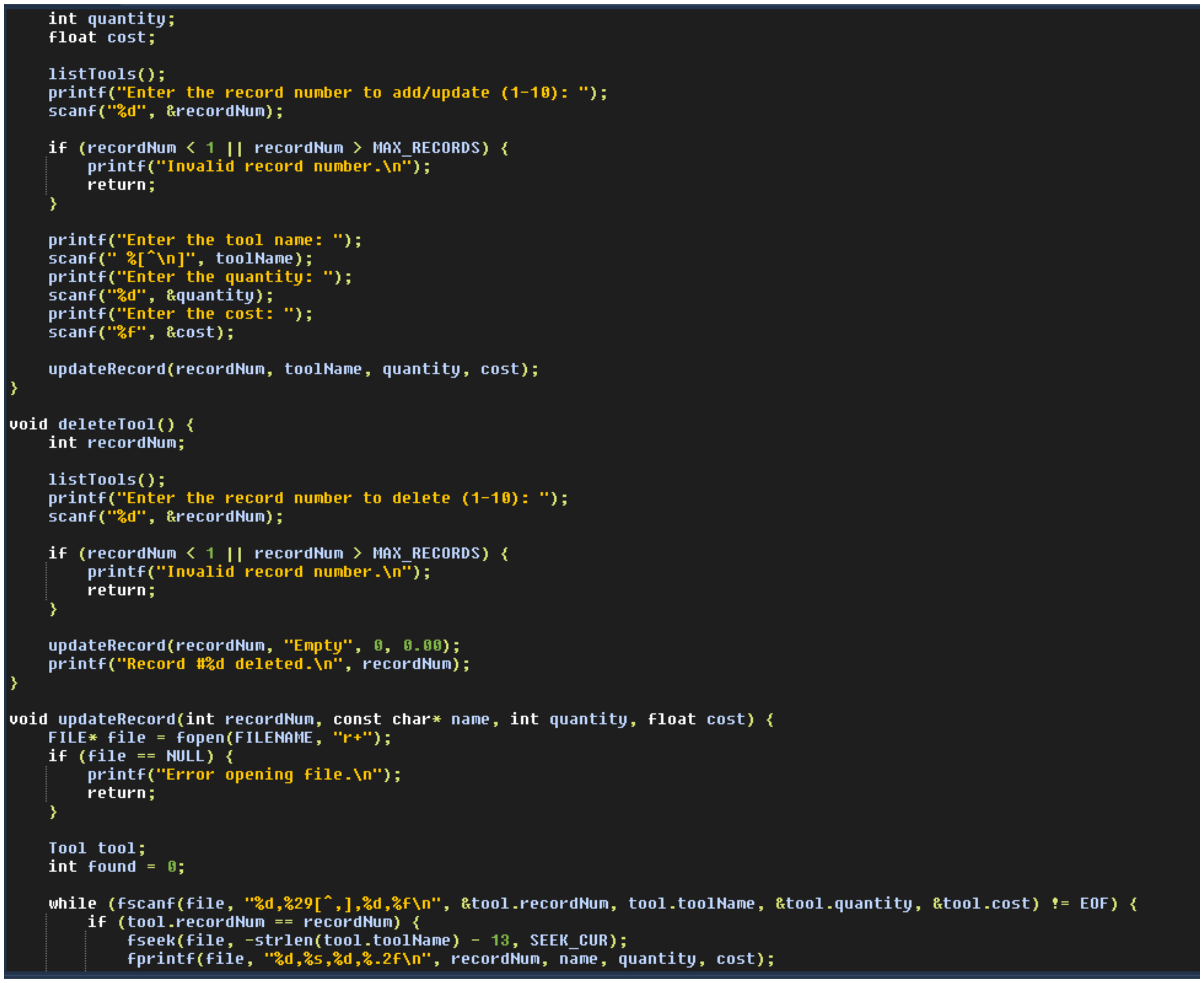
3. You're the owner of a hardware store and need to keep an inventory that can tell you what tools you have, how many you have and the cost of each one. Write a program that initializes the file "hardware.txt" to 10 empty records, lets you input the data concerning each tool, enables you to list all your tools, lets you delete a record for a tool that you no longer have and lets you update any information in the file. The tool identification number should be the record number. Use the following information to start your file.



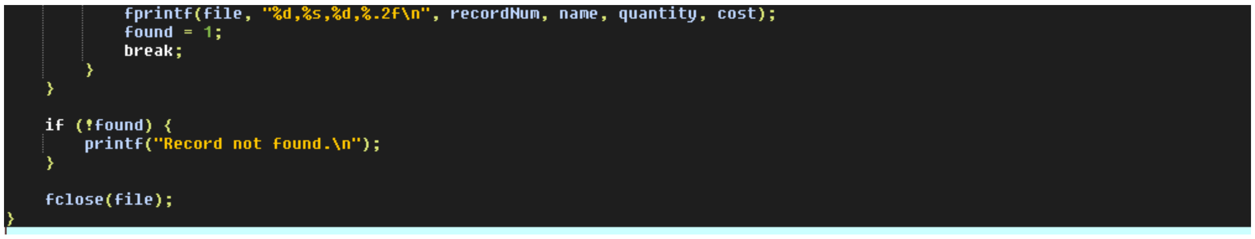
```c
// . You're the owner of a hardware store and need to keep an inventory that can tell you what tools
// you have, how many you have and the cost of each one. Write a program that initializes the file
// "hardware.txt" to 10 empty records, lets you input the data concerning each tool, enables you
// to list all your tools, lets you delete a record for a tool that you no longer have and lets you
// update any information in the file. The tool identification number should be the record
// number. Use the following information to start your file:


// Name : Mustafa Ahmed Siddiqui
// Class : BCIT F
// Roll No : CT-261

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

#define FILENAME "hardware.txt"
#define MAX_RECORDS 10

typedef struct {
    int recordNum;
    char toolName[30];
    int quantity;
    float cost;
} Tool;

void initializeFile();
void listTools();
void addOrUpdateTool();
void deleteTool();
void updateRecord(int recordNum, const char* name, int quantity, float cost);

int main() {
    int choice;

    initializeFile();

    do {
        printf("\n--- Hardware Store Inventory Management ---\n");
        printf("1. List all tools\n");
        printf("2. Add/Update a tool\n");
        printf("3. Delete a tool\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                listTools();
                break;
            case 2:
                addOrUpdateTool();
```



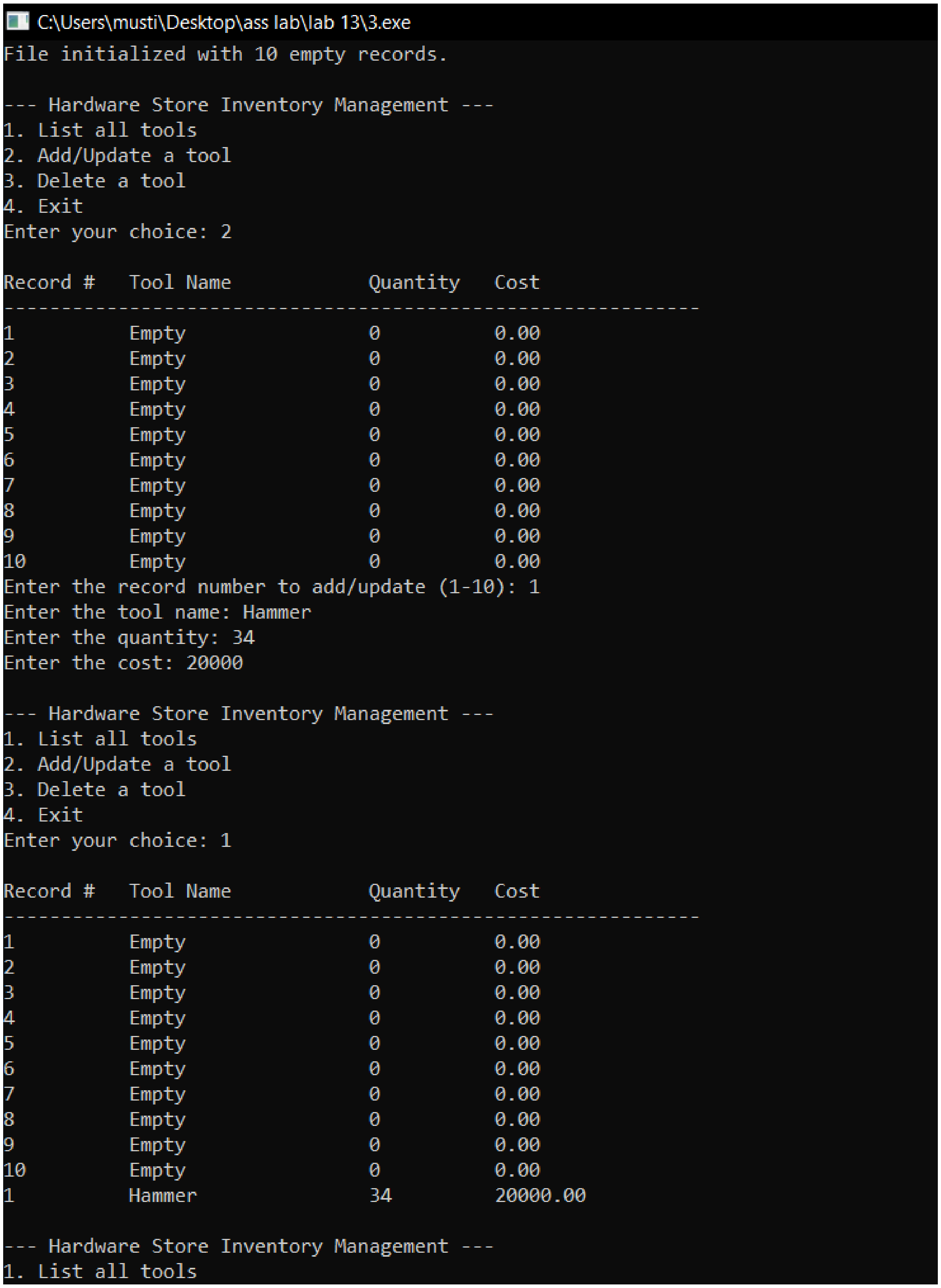
```c
                addOrUpdateTool();
                break;
            case 3:
                deleteTool();
                break;
            case 4:
                printf("Exiting program.\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 4);

    return 0;
}

void initializeFile() {
    FILE* file = fopen(FILENAME, "r");
    if (file == NULL) {
        file = fopen(FILENAME, "w");
        for (int i = 1; i <= MAX_RECORDS; i++) {
            fprintf(file, "%d,Empty,0,0.00\n", i);
        }
        fclose(file);
        printf("File initialized with 10 empty records.\n");
    } else {
        fclose(file);
    }
}

void listTools() {
    FILE* file = fopen(FILENAME, "r");
    if (file == NULL) {
        printf("Error opening file.\n");
        return;
    }

    Tool tool;
    printf("\n%-10s %-20s %-10s %-10s\n", "Record #", "Tool Name", "Quantity", "Cost");
    printf("------------------------------------------------------------\n");

    while (fscanf(file, "%d,%29[^,],%d,%f\n", &tool.recordNum, tool.toolName, &tool.quantity, &tool.cost) != EOF) {
        printf("%-10d %-20s %-10d %-10.2f\n", tool.recordNum, tool.toolName, tool.quantity, tool.cost);
    }

    fclose(file);
}

void addOrUpdateTool() {
    int recordNum;
    char toolName[30];
    int quantity;
```



```c
    int quantity;
    float cost;

    listTools();
    printf("Enter the record number to add/update (1-10): ");
    scanf("%d", &recordNum);

    if (recordNum < 1 || recordNum > MAX_RECORDS) {
        printf("Invalid record number.\n");
        return;
    }

    printf("Enter the tool name: ");
    scanf(" %[^\n]", toolName);
    printf("Enter the quantity: ");
    scanf("%d", &quantity);
    printf("Enter the cost: ");
    scanf("%f", &cost);

    updateRecord(recordNum, toolName, quantity, cost);
}

void deleteTool() {
    int recordNum;

    listTools();
    printf("Enter the record number to delete (1-10): ");
    scanf("%d", &recordNum);

    if (recordNum < 1 || recordNum > MAX_RECORDS) {
        printf("Invalid record number.\n");
        return;
    }

    updateRecord(recordNum, "Empty", 0, 0.00);
    printf("Record #%d deleted.\n", recordNum);
}

void updateRecord(int recordNum, const char* name, int quantity, float cost) {
    FILE* file = fopen(FILENAME, "r+");
    if (file == NULL) {
        printf("Error opening file.\n");
        return;
    }

    Tool tool;
    int found = 0;

    while (fscanf(file, "%d,%29[^,],%d,%f\n", &tool.recordNum, tool.toolName, &tool.quantity, &tool.cost) != EOF) {
        if (tool.recordNum == recordNum) {
            fseek(file, -strlen(tool.toolName) - 13, SEEK_CUR);
            fprintf(file, "%d,%s,%d,%.2f\n", recordNum, name, quantity, cost);
```



```c
            fprintf(file, "%d,%s,%d,%.2f\n", recordNum, name, quantity, cost);
            found = 1;
            break;
        }
    }

    if (!found) {
        printf("Record not found.\n");
    }

    fclose(file);
}
```



C:\Users\musti\Desktop\ass lab\lab 13\3.exe

```
File initialized with 10 empty records.

--- Hardware Store Inventory Management ---
1. List all tools
2. Add/Update a tool
3. Delete a tool
4. Exit
Enter your choice: 2

Record #   Tool Name            Quantity   Cost
------------------------------------------------------------
1          Empty                0          0.00
2          Empty                0          0.00
3          Empty                0          0.00
4          Empty                0          0.00
5          Empty                0          0.00
6          Empty                0          0.00
7          Empty                0          0.00
8          Empty                0          0.00
9          Empty                0          0.00
10         Empty                0          0.00
Enter the record number to add/update (1-10): 1
Enter the tool name: Hammer
Enter the quantity: 34
Enter the cost: 20000

--- Hardware Store Inventory Management ---
1. List all tools
2. Add/Update a tool
3. Delete a tool
4. Exit
Enter your choice: 1

Record #   Tool Name            Quantity   Cost
------------------------------------------------------------
1          Empty                0          0.00
2          Empty                0          0.00
3          Empty                0          0.00
4          Empty                0          0.00
5          Empty                0          0.00
6          Empty                0          0.00
7          Empty                0          0.00
8          Empty                0          0.00
9          Empty                0          0.00
10         Empty                0          0.00
1          Hammer               34         20000.00

--- Hardware Store Inventory Management ---
1. List all tools
```