#include <stdio.h>

#include <string.h>

// Define the Student structure

struct Student {

int ID;

char name[50];

char gender;

float quiz1, quiz2, midtermScore, finalScore, totalScore;

};

// Global variables

struct Student students[20];

int studentCount = 0;

// Function prototypes

void addStudent();

void deleteStudent();

void updateStudent();

void viewStudents();

void calculateAverage();

void findMaxScore();

void findMinScore();

void findStudentByID();

void sortRecords();

void menu();

// Main function

int main() {

int choice;

do {

menu();

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1: addStudent(); break;

case 2: deleteStudent(); break;

case 3: updateStudent(); break;

case 4: viewStudents(); break;

case 5: calculateAverage(); break;

case 6: findMaxScore(); break;

case 7: findMinScore(); break;

case 8: findStudentByID(); break;

case 9: sortRecords(); break;

case 0: printf("Exiting the program...\n"); break;

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

}

} while (choice != 0);

return 0;

}

// Menu function

void menu() {

printf("\n-------------------\n");

printf(" Menu \n");

printf("-------------------\n");

printf("1. Add student records\n");

printf("2. Delete student records\n");

printf("3. Update student records\n");

printf("4. View all student records\n");

printf("5. Calculate an average of a selected student's scores\n");

printf("6. Show student who gets the max total score\n");

printf("7. Show student who gets the min total score\n");

printf("8. Find student by ID\n");

printf("9. Sort records by total scores\n");

printf("0. Exit\n");

}

// Function to add a student

void addStudent() {

if (studentCount >= 20) {

printf("Student limit reached! Cannot add more records.\n");

return;

}

struct Student s;

printf("Enter Student ID: ");

scanf("%d", &s.ID);

printf("Enter Name: ");

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

printf("Enter Gender (M/F): ");

scanf(" %c", &s.gender);

printf("Enter Quiz 1 Score: ");

scanf("%f", &s.quiz1);

printf("Enter Quiz 2 Score: ");

scanf("%f", &s.quiz2);

printf("Enter Mid-term Score: ");

scanf("%f", &s.midtermScore);

printf("Enter Final Score: ");

scanf("%f", &s.finalScore);

s.totalScore = s.quiz1 + s.quiz2 + s.midtermScore + s.finalScore;

students[studentCount++] = s;

printf("Student record added successfully!\n");

}

// Function to delete a student

void deleteStudent() {

int id, found = 0;

printf("Enter the ID of the student to delete: ");

scanf("%d", &id);

for (int i = 0; i < studentCount; i++) {

if (students[i].ID == id) {

for (int j = i; j < studentCount - 1; j++) {

students[j] = students[j + 1];

}

studentCount--;

found = 1;

printf("Student record deleted successfully!\n");

break;

}

}

if (!found) {

printf("Student with ID %d not found.\n", id);

}

}

// Function to update a student record

void updateStudent() {

int id, found = 0;

printf("Enter the ID of the student to update: ");

scanf("%d", &id);

for (int i = 0; i < studentCount; i++) {

if (students[i].ID == id) {

printf("Enter new Quiz 1 Score: ");

scanf("%f", &students[i].quiz1);

printf("Enter new Quiz 2 Score: ");

scanf("%f", &students[i].quiz2);

printf("Enter new Mid-term Score: ");

scanf("%f", &students[i].midtermScore);

printf("Enter new Final Score: ");

scanf("%f", &students[i].finalScore);

students[i].totalScore = students[i].quiz1 + students[i].quiz2 + students[i].midtermScore + students[i].finalScore;

found = 1;

printf("Student record updated successfully!\n");

break;

}

}

if (!found) {

printf("Student with ID %d not found.\n");

}

}

// Function to view all student records

void viewStudents() {

if (studentCount == 0) {

printf("No student records found.\n");

return;

}

printf("\n%-5s %-20s %-5s %-6s %-6s %-8s %-8s %-6s\n", "ID", "Name", "Sex", "Quiz1", "Quiz2", "Midterm", "Final", "Total");

printf("----------------------------------------------------------------\n");

for (int i = 0; i < studentCount; i++) {

printf("%-5d %-20s %-5c %-6.2f %-6.2f %-8.2f %-8.2f %-6.2f\n",

students[i].ID, students[i].name, students[i].gender,

students[i].quiz1, students[i].quiz2, students[i].midtermScore,

students[i].finalScore, students[i].totalScore);

}

}

// Function to calculate the average score of a student

void calculateAverage() {

int id, found = 0;

printf("Enter the ID of the student: ");

scanf("%d", &id);

for (int i = 0; i < studentCount; i++) {

if (students[i].ID == id) {

float average = students[i].totalScore / 4.0;

printf("Average score of student ID %d: %.2f\n", id, average);

found = 1;

break;

}

}

if (!found) {

printf("Student with ID %d not found.\n", id);

}

}

// Function to find the student with the maximum score

void findMaxScore() {

if (studentCount == 0) {

printf("No student records found.\n");

return;

}

struct Student max = students[0];

for (int i = 1; i < studentCount; i++) {

if (students[i].totalScore > max.totalScore) {

max = students[i];

}

}

printf("Student with max score: %s (ID: %d), Total: %.2f\n", max.name, max.ID, max.totalScore);

}

// Function to find the student with the minimum score

void findMinScore() {

if (studentCount == 0) {

printf("No student records found.\n");

return;

}

struct Student min = students[0];

for (int i = 1; i < studentCount; i++) {

if (students[i].totalScore < min.totalScore) {

min = students[i];

}

}

printf("Student with min score: %s (ID: %d), Total: %.2f\n", min.name, min.ID, min.totalScore);

}

// Function to find a student by ID

void findStudentByID() {

int id, found = 0;

printf("Enter the ID of the student: ");

scanf("%d", &id);

for (int i = 0; i < studentCount; i++) {

if (students[i].ID == id) {

printf("Student found: %s (ID: %d), Total: %.2f\n",

students[i].name, students[i].ID, students[i].totalScore);

found = 1;

break;

}

}

if (!found) {

printf("Student with ID %d not found.\n", id);

}

}

// Function to sort records by total score

void sortRecords() {

if (studentCount == 0) {

printf("No student records to sort.\n");

return;

}

for (int i = 0; i < studentCount - 1; i++) {

for (int j = 0; j < studentCount - i - 1; j++) {

if (students[j].totalScore < students[j + 1].totalScore) {

struct Student temp = students[j];

students[j] = students[j + 1];

students[j + 1] = temp;

}

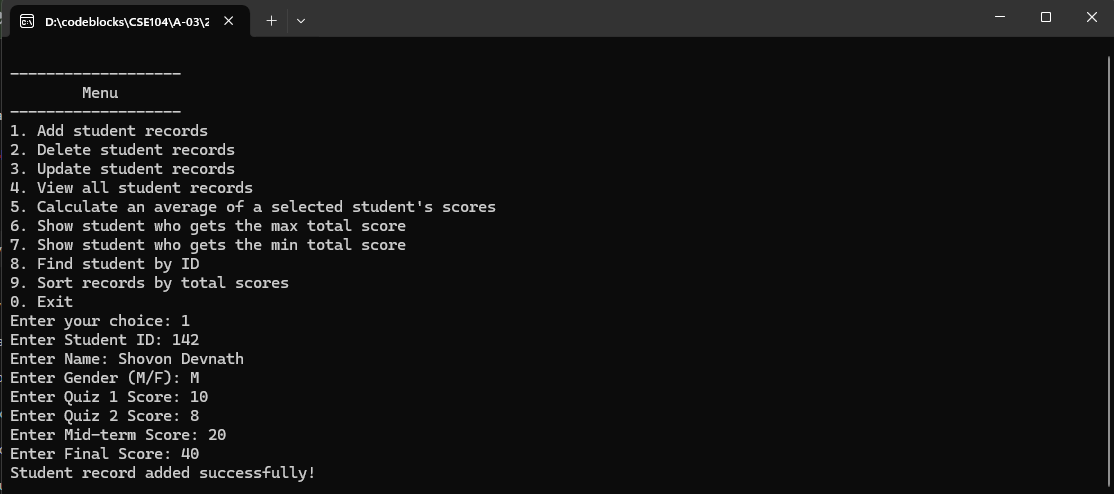
}

}

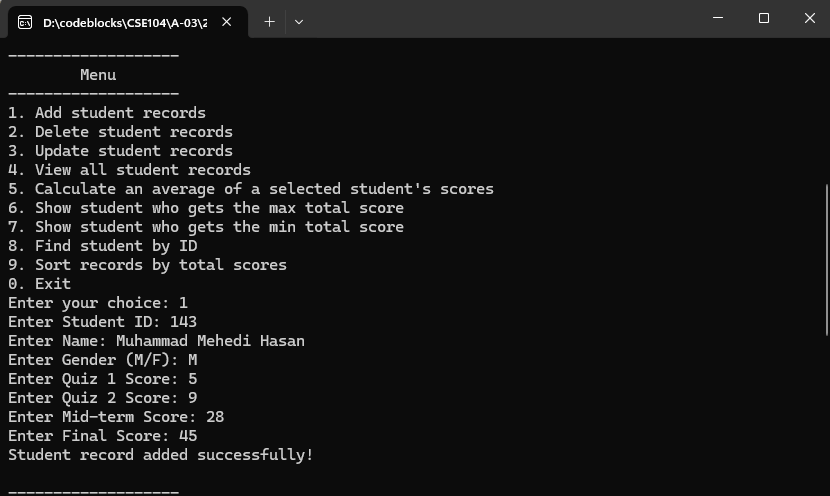
printf("Records sorted by total scores in descending order.\n");

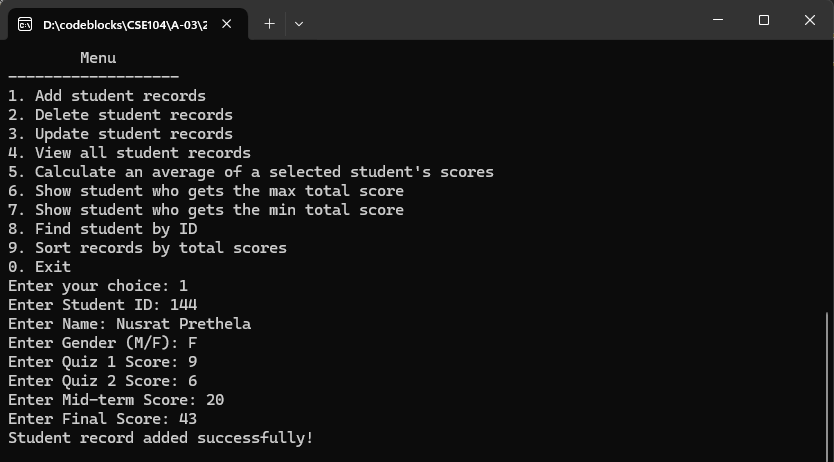
}   
  
  
Output:

Add student records

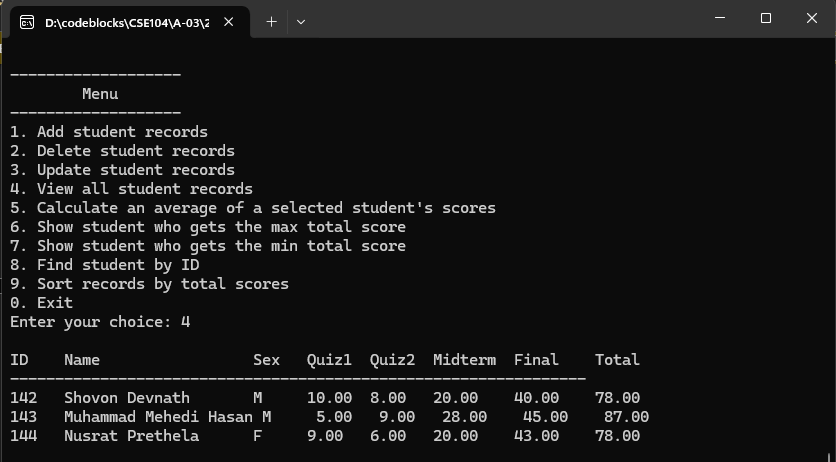


Added another two student records

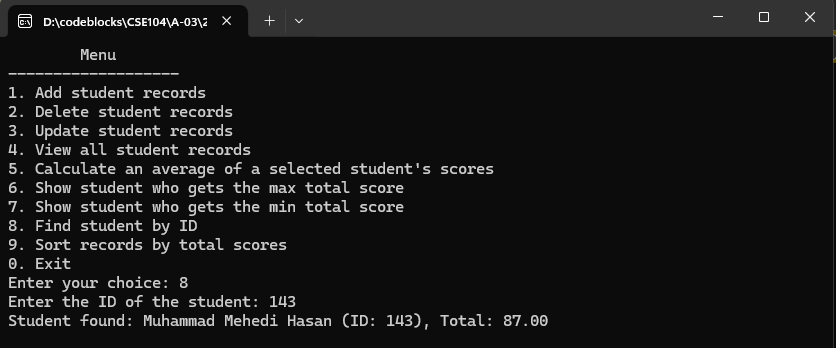




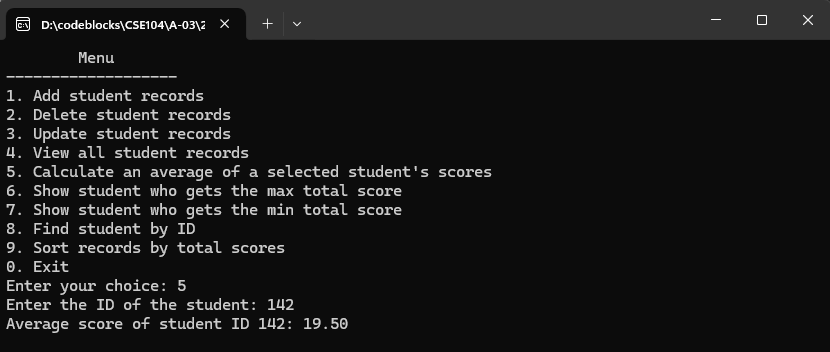
View All Student Records



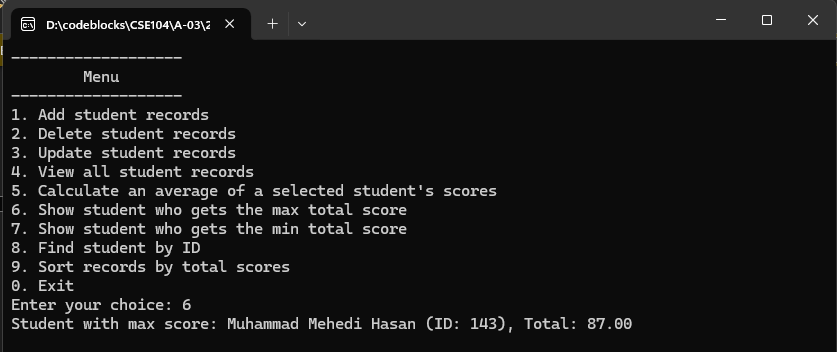
Find Student by ID



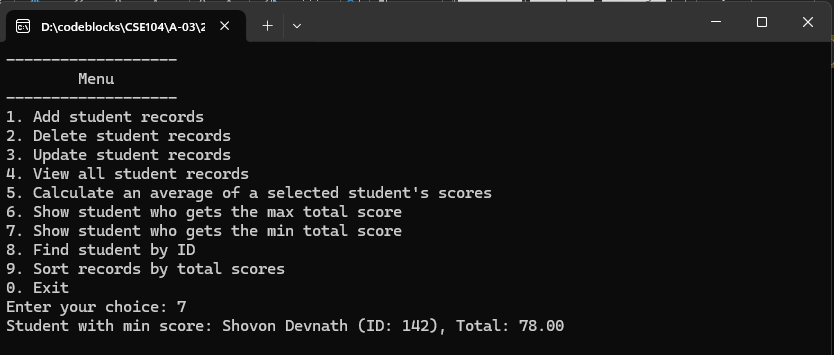
Calculate Average Score of a Student



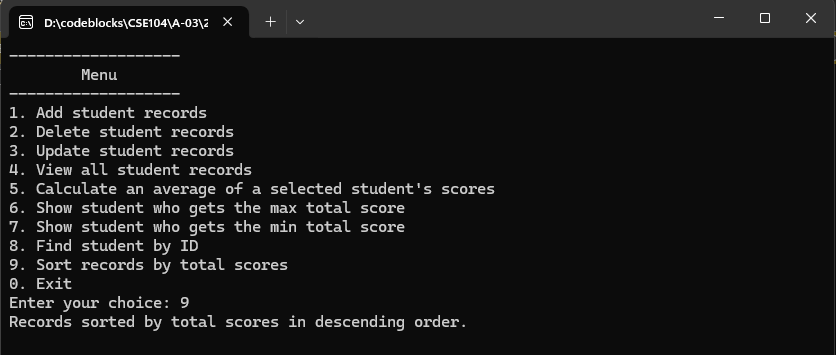
Show Student with the Max Total Score



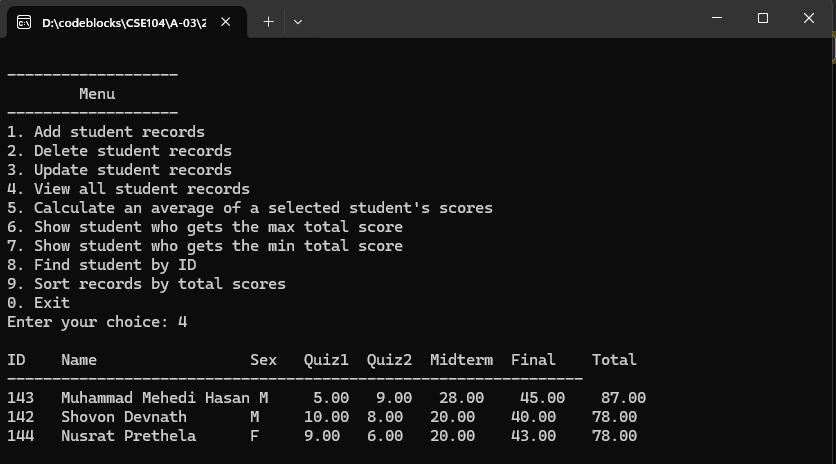
Show Student with the Min Total Score



Sort Records by Total Scores

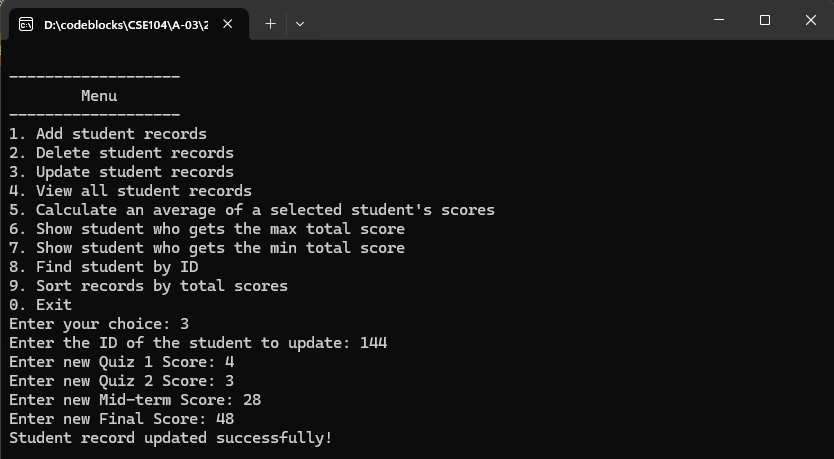


View After Sorting:

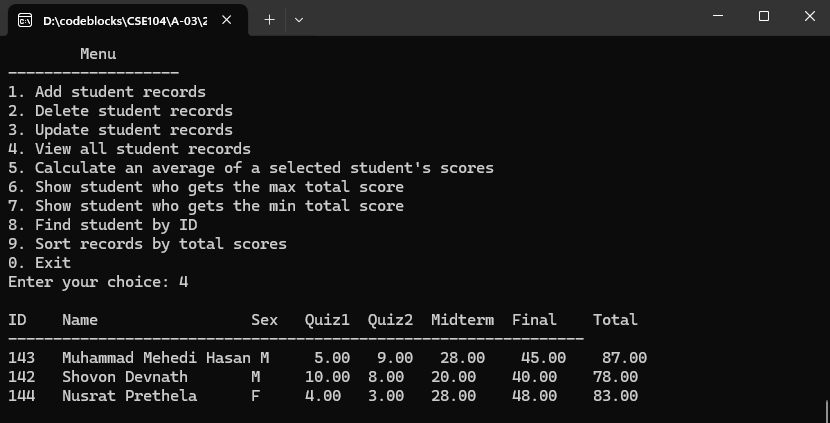


Update Student Records

Input:



Updated Records:



Delete a Student

