#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_STUDENTS 100

#define FILENAME "students.txt"

struct Student {

int rollNo;

char name[50];

char course[50];

int marks;

};

void saveRecords(struct Student students[], int count) {

FILE \*file = fopen(FILENAME, "w");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

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

fprintf(file, "%d, %s, %s, %d\n", students[i].rollNo, students[i].name, students[i].course, students[i].marks);

}

fclose(file);

printf("Record saved in \"%s\"\n", FILENAME);

}

struct Student\* searchStudent(struct Student students[], int count, int rollNo) {

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

if (students[i].rollNo == rollNo) {

return &students[i];

}

}

return NULL;

}

int main() {

struct Student students[MAX\_STUDENTS];

int count;

printf("Enter number of students: ");

scanf("%d", &count);

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

printf("Student %d: ", i + 1);

scanf("%d, %49[^,], %49[^,], %d", &students[i].rollNo, students[i].name, students[i].course, &students[i].marks);

}

saveRecords(students, count);

int rollNo;

printf("Enter Roll No to search: ");

scanf("%d", &rollNo);

struct Student \*foundStudent = searchStudent(students, count, rollNo);

if (foundStudent) {

printf("Student Found: %s, %s, Marks: %d\n", foundStudent->name, foundStudent->course, foundStudent->marks);

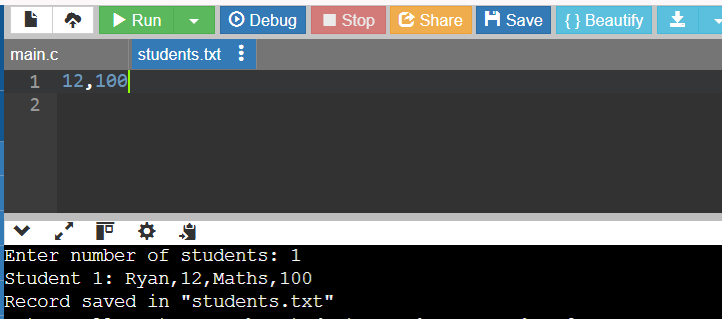
} else {

printf("Student with Roll No %d not found.\n", rollNo);

}

return 0;

}



#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_ORDERS 100

#define FILENAME "orders.txt"

struct Order {

int orderId;

char customerName[50];

char product[50];

float amount;

};

void saveOrders(struct Order orders[], int count) {

FILE \*file = fopen(FILENAME, "w");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

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

fprintf(file, "%d, %s, %s, ₹%.2f\n", orders[i].orderId, orders[i].customerName, orders[i].product, orders[i].amount);

}

fclose(file);

printf("Record saved in \"%s\"\n", FILENAME);

}

struct Order\* searchOrder(struct Order orders[], int count, int orderId) {

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

if (orders[i].orderId == orderId) {

return &orders[i];

}

}

return NULL;

}

int main() {

struct Order orders[MAX\_ORDERS];

int count;

printf("Enter number of orders: ");

scanf("%d", &count);

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

printf("Order %d: ", i + 1);

scanf("%d, %49[^,], %49[^,], %f", &orders[i].orderId, orders[i].customerName, orders[i].product, &orders[i].amount);

}

saveOrders(orders, count);

int orderId;

printf("Enter Order ID to search: ");

scanf("%d", &orderId);

struct Order \*foundOrder = searchOrder(orders, count, orderId);

if (foundOrder) {

printf("Order Found: %s - %s - ₹%.2f\n", foundOrder->customerName, foundOrder->product, foundOrder->amount);

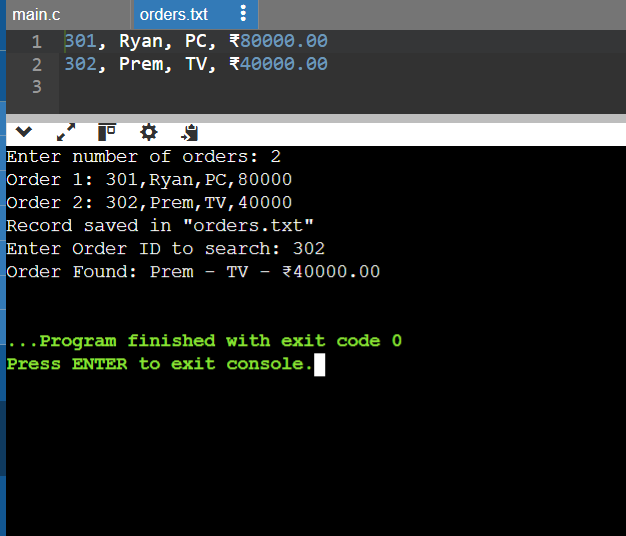
} else {

printf("Order with ID %d not found.\n", orderId);

}

return 0;

}



#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_CITIES 100

#define FILENAME "weather\_data.txt"

struct WeatherRecord {

char city[50];

float temperature; // in °C

float humidity; // in %

};

void saveWeatherData(struct WeatherRecord records[], int count) {

FILE \*file = fopen(FILENAME, "w");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

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

fprintf(file, "%s, %.1f°C, %.1f%%\n", records[i].city, records[i].temperature, records[i].humidity);

}

fclose(file);

printf("Data saved in \"%s\"\n", FILENAME);

}

struct WeatherRecord\* searchCityWeather(struct WeatherRecord records[], int count, const char \*city) {

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

if (strcmp(records[i].city, city) == 0) {

return &records[i];

}

}

return NULL;

}

int main() {

struct WeatherRecord records[MAX\_CITIES];

int count;

printf("Enter number of cities: ");

scanf("%d", &count);

getchar(); // To consume the newline character after the number input

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

printf("City %d: ", i + 1);

scanf("%49[^,], %f°C, %f%%", records[i].city, &records[i].temperature, &records[i].humidity);

getchar(); // To consume the newline character after each input

}

saveWeatherData(records, count);

char city[50];

printf("Enter city name to search: ");

fgets(city, sizeof(city), stdin);

city[strcspn(city, "\n")] = 0; // Remove newline character from input

struct WeatherRecord \*foundRecord = searchCityWeather(records, count, city);

if (foundRecord) {

printf("Weather in %s: %.1f°C, %.1f%% Humidity\n", foundRecord->city, foundRecord->temperature, foundRecord->humidity);

} else {

printf("Weather data for %s not found.\n", city);

}

return 0;

}

