**LAB 11**

* **Question 1:**
* Source code:

#include <stdio.h>

struct employee{

char name[20];

int id;

float salary;

};

int main()

{

struct employee e[3];

int i;

float sum=0;

for (i=0;i<3;i++)

{

printf("Employee %d\n",i+1);

printf("Enter employee name: ");

scanf("%s",e[i].name);

printf("Enter employee id: ");

scanf("%d",&e[i].id);

printf("Enter employee salary: ");

scanf("%f",&e[i].salary);

}

for (i=0;i<3;i++)

{

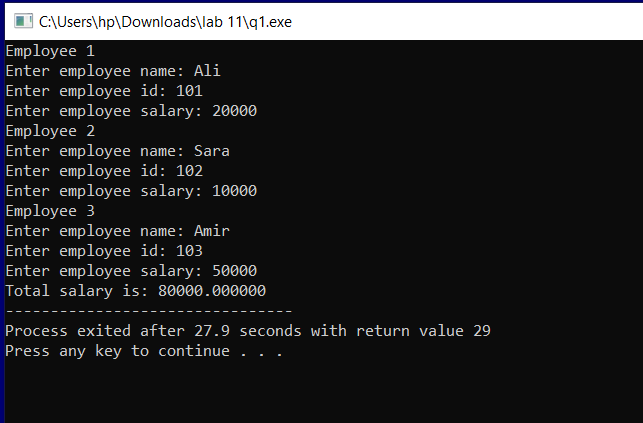
sum+=e[i].salary;

}

printf("Total salary is: %f",sum);

}

* Output:



* **Question 2:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

struct employee{

char name[20];

int id;

float salary;

};

int main()

{

struct employee e[3];

struct employee temp;

int i;

float sum=0;

FILE \*file;

file=fopen("Student\_data.txt","wb");

if(file==NULL)

{

printf("Error in opening file");

return 1;

}

for (i=0;i<3;i++)

{

printf("Employee %d\n",i+1);

printf("Enter employee name: ");

scanf("%s",e[i].name);

printf("Enter employee id: ");

scanf("%d",&e[i].id);

printf("Enter employee salary: ");

scanf("%f",&e[i].salary);

}

fwrite(e, sizeof(struct employee), 3, file);

printf("\nEmployee record saved successfully.\n");

fclose(file);

file = fopen("Student\_data.txt", "rb");

if (file == NULL) {

printf("Error in opening file for reading.\n");

return 1;

}

while (fread(&temp, sizeof(struct employee), 1, file))

{

sum+=temp.salary;

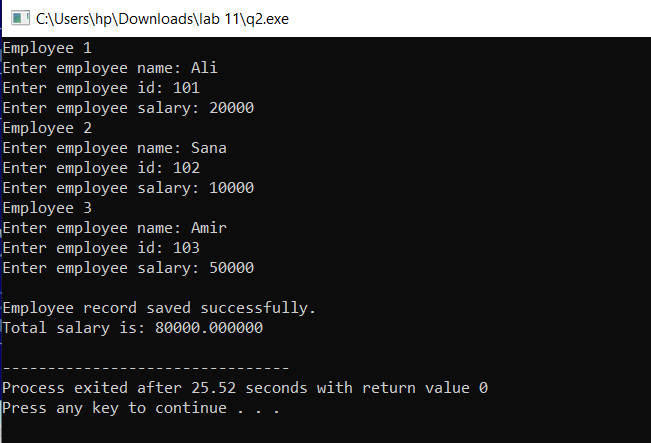
}

printf("Total salary is: %f\n",sum);

fclose(file);

}

* Output:



* **Question 3:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Course

{

char name[20];

char code[10];

char instructor[50];

};

struct Student

{

char name[20];

char roll\_no[10];

char courses[10][50];

int course\_count;

};

void addCourses();

void addStudents();

void display\_course();

void report();

void update();

int main()

{

int choice;

while (choice!=6)

{

printf("\n--- Student Course Enrollment System ---\n");

printf("1. Add Courses\n");

printf("2. Add Students\n");

printf("3. Display Courses Enrolled by a Student\n");

printf("4. Generate Enrollment Report\n");

printf("5. Modify Course Enrollment\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

addCourses();

break;

case 2:

addStudents();

break;

case 3:

display\_course();

break;

case 4:

report();

break;

case 5:

update();

break;

case 6:

printf("Exiting program---");

break;

default:

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

}

}

return 0;

}

void addCourses()

{

struct Course c[10];

int count,i;

FILE \*file = fopen("courses.txt", "wb");

if (file == NULL)

{

printf("Error opening file for writing courses.\n");

return;

}

printf("Enter the number of courses to add (max 10): ");

scanf("%d", &count);

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

{

printf("Course %d:\n", i + 1);

printf("Enter course name: ");

scanf("%s", c[i].name);

printf("Enter course code: ");

scanf("%s", c[i].code);

printf("Enter course instructor: ");

scanf("%s", c[i].instructor);

}

fwrite(c, sizeof(struct Course), count, file);

printf("Courses saved successfully.\n");

fclose(file);

}

void addStudents()

{

struct Student s[10];

int count,i,j;

FILE \*file = fopen("students.txt", "ab");

if (file == NULL)

{

printf("Error opening file for writing students.\n");

return;

}

printf("Enter the number of students to add (max 10): ");

scanf("%d", &count);

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

{

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

printf("Enter student name: ");

scanf("%s", s[i].name);

printf("Enter roll number: ");

scanf("%s", s[i].roll\_no);

printf("Enter the number of courses enrolled (max 10): ");

scanf("%d", &s[i].course\_count);

for (j = 0; j < s[i].course\_count; j++)

{

printf("Enter course %d name: ", j + 1);

scanf("%s", s[i].courses[j]);

}

}

fwrite(s, sizeof(struct Student), count, file);

printf("Students saved successfully.\n");

fclose(file);

}

void display\_course()

{

struct Student s;

char roll\_no[10];

int found = 0,i;

FILE \*file = fopen("students.txt", "rb");

if (file == NULL)

{

printf("Error opening file for reading students.\n");

return;

}

printf("Enter roll number of the student: ");

scanf("%s", roll\_no);

while (fread(&s, sizeof(struct Student), 1, file))

{

if (strcmp(s.roll\_no, roll\_no) == 0)

{

printf("Student Name: %s\n", s.name);

printf("Roll Number: %s\n", s.roll\_no);

printf("Courses Enrolled:\n");

for (i = 0; i < s.course\_count; i++)

{

printf("- %s\n", s.courses[i]);

}

found = 1;

break;

}

}

if (!found) {

printf("Student with roll number %s not found.\n", roll\_no);

}

fclose(file);

}

void report()

{

struct Student s;

int count[10] = {0},i,j;

char course\_names[10][50] = {"PF", "Calc", "FE", "AP", "OOP", "DLD"};

FILE \*file = fopen("students.txt", "rb");

if (file == NULL)

{

printf("Error opening file for reading students.\n");

return;

}

while (fread(&s, sizeof(struct Student), 1, file))

{

for (i = 0; i < s.course\_count; i++)

{

for (j = 0; j < 10; j++)

{

if (strcmp(s.courses[i], course\_names[j]) == 0)

{

count[j]++;

}

}

}

}

fclose(file);

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

for (i = 0; i < 10; i++) {

if (strlen(course\_names[i]) > 0)

{

printf("%s: %d students\n", course\_names[i], count[i]);

}

}

}

void update()

{

struct Student s[10];

char roll\_no[10];

int found = 0, student\_count = 0,i,j;

FILE \*file = fopen("students.bin", "rb");

if (file == NULL)

{

printf("Error opening file for reading students.\n");

return;

}

while (fread(&s[student\_count], sizeof(struct Student), 1, file))

{

student\_count++;

}

fclose(file);

printf("Enter roll number of the student to modify: ");

scanf("%s", roll\_no);

for (i = 0; i < student\_count; i++)

{

if (strcmp(s[i].roll\_no, roll\_no) == 0)

{

found = 1;

printf("Modifying enrollment for %s:\n", s[i].name);

printf("Enter the new number of courses (max 10): ");

scanf("%d", &s[i].course\_count);

for (j = 0; j < s[i].course\_count; j++)

{

printf("Enter course %d name: ", j + 1);

scanf("%s", s[i].courses[j]);

}

break;

}

}

if (!found)

{

printf("Student with roll number %s not found.\n", roll\_no);

return;

}

file = fopen("students.bin", "wb");

if (file == NULL)

{

printf("Error opening file for updating students.\n");

return;

}

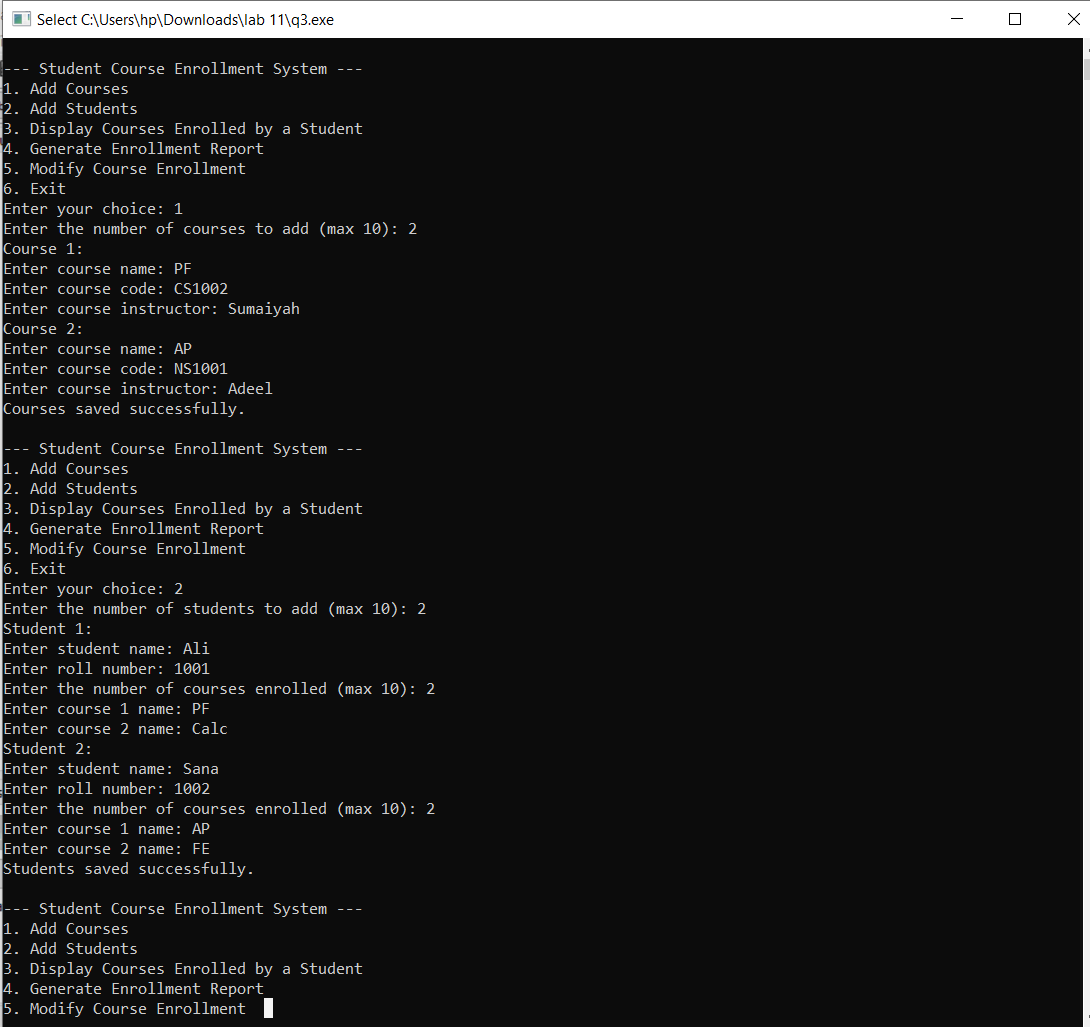
fwrite(s, sizeof(struct Student), student\_count, file);

printf("Student enrollment updated successfully.\n");

fclose(file);

}

* Output:



* **Question 4:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_PASSENGERS 1000

#define MAX\_LINE\_LENGTH 512

typedef struct {

int passengerId;

int survived;

int pclass;

char name[100];

char sex[10];

float age;

int sibSp;

int parch;

char ticket[20];

float fare;

char cabin[20];

char embarked[5];

} Passenger;

int loadPassengers(const char \*filename, Passenger passengers[]);

float calculateAverageAge(Passenger passengers[], int size);

void calculateSurvivalRateBySex(Passenger passengers[], int size);

void displaySurvivors(Passenger passengers[], int size);

void displayMissingCabinDetails(Passenger passengers[], int size);

int main()

{

Passenger passengers[MAX\_PASSENGERS];

int passengerCount;

passengerCount = loadPassengers("train.csv", passengers);

if (passengerCount == -1)

{

printf("Error reading the file.\n");

return 1;

}

float average\_age = calculateAverageAge(passengers, passengerCount);

printf("Average Age of Passengers: %.2f\n", average\_age);

calculateSurvivalRateBySex(passengers, passengerCount);

printf("\nList of Survivors:\n");

displaySurvivors(passengers, passengerCount);

printf("\nPassengers with Missing Cabin Details:\n");

displayMissingCabinDetails(passengers, passengerCount);

return 0;

}

void parseCSVLine(char \*line, char \*fields[], int maxFields)

{

int fieldIndex = 0, i;

char \*start = line, \*ptr;

int inQuotes = 0;

for (ptr = line; \*ptr; ptr++)

{

if (\*ptr == '"')

{

inQuotes = !inQuotes;

} else if (\*ptr == ',' && !inQuotes)

{

\*ptr = '\0';

fields[fieldIndex++] = start;

start = ptr + 1;

}

}

fields[fieldIndex++] = start;

for (i = fieldIndex; i < maxFields; i++) fields[i] = NULL;

}

int loadPassengers(const char \*filename, Passenger passengers[])

{

FILE \*file = fopen(filename, "r");

if (!file) return -1;

char line[MAX\_LINE\_LENGTH];

int index = 0;

fgets(line, sizeof(line), file);

while (fgets(line, sizeof(line), file))

{

char \*fields[12];

parseCSVLine(line, fields, 12);

passengers[index].passengerId = atoi(fields[0]);

passengers[index].survived = atoi(fields[1]);

passengers[index].pclass = atoi(fields[2]);

strncpy(passengers[index].name, fields[3], sizeof(passengers[index].name) - 1);

strncpy(passengers[index].sex, fields[4], sizeof(passengers[index].sex) - 1);

passengers[index].age = (fields[5] && \*fields[5]) ? atof(fields[5]) : 0; // Missing age as 0

passengers[index].sibSp = atoi(fields[6]);

passengers[index].parch = atoi(fields[7]);

strncpy(passengers[index].ticket, fields[8], sizeof(passengers[index].ticket) - 1);

passengers[index].fare = (fields[9] && \*fields[9]) ? atof(fields[9]) : 0;

strncpy(passengers[index].cabin, fields[10] ? fields[10] : "", sizeof(passengers[index].cabin) - 1);

strncpy(passengers[index].embarked, fields[11] ? fields[11] : "", sizeof(passengers[index].embarked) - 1);

index++;

}

fclose(file);

return index;

}

float calculateAverageAge(Passenger passengers[], int size)

{

float totalAge = 0;

int count = 0, i;

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

{

if (passengers[i].age > 0)

{

totalAge += passengers[i].age;

count++;

}

}

return count > 0 ? totalAge / count : 0;

}

void calculateSurvivalRateBySex(Passenger passengers[], int size)

{

int maleSurvivors = 0, femaleSurvivors = 0;

int maleCount = 0, femaleCount = 0, i;

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

{

if (strcmp(passengers[i].sex, "male") == 0)

{

maleCount++;

if (passengers[i].survived) maleSurvivors++;

} else if (strcmp(passengers[i].sex, "female") == 0)

{

femaleCount++;

if (passengers[i].survived) femaleSurvivors++;

}

}

printf("\nSurvival Rate by Sex:\n");

printf("Male Survival Rate: %.2f%%\n", (maleCount > 0) ? (maleSurvivors \* 100.0 / maleCount) : 0);

printf("Female Survival Rate: %.2f%%\n", (femaleCount > 0) ? (femaleSurvivors \* 100.0 / femaleCount) : 0);

}

void displaySurvivors(Passenger passengers[], int size)

{

int count = 1, i;

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

{

if (passengers[i].survived)

{

printf("%d. Name: %s, ", count++, passengers[i].name);

if (passengers[i].age > 0)

printf("Age: %.1f, ", passengers[i].age);

printf("Class: %d\n", passengers[i].pclass);

}

}

}

void displayMissingCabinDetails(Passenger passengers[], int size)

{

int count = 1, i;

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

{

if (strlen(passengers[i].cabin) == 0 || strcmp(passengers[i].cabin, " ") == 0)

{

printf("%d. Name: %s, ", count++, passengers[i].name);

if (passengers[i].age > 0)

printf("Age: %.1f, ", passengers[i].age);

printf("Class: %d\n", passengers[i].pclass);

}

}

}

* Output:

