**1. Student Grade System**

**Scenario/Question**

You are building a student grading system. For each student, store their name, roll number, and 3 subject marks. Calculate the average and display the result.

**Code**

#include <stdio.h>

struct Student {

char name[50];

int rollNo;

float marks[3];

};

int main() {

struct Student s;

float sum = 0;

printf("Enter name and roll number: ");

scanf("%s %d", s.name, &s.rollNo);

printf("Enter marks in 3 subjects:\n");

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

scanf("%f", &s.marks[i]);

sum += s.marks[i];

}

printf("Student: %s\nAverage Marks: %.2f\n", s.name, sum / 3);

return 0;

}

**Sample Output**

Enter name and roll number: Ravi 101

Enter marks in 3 subjects:

78 82 90

Student: Ravi

Average Marks: 83.33

**2. Bank Account Management**

**Scenario/Question**

Create a program that stores bank account details and performs a deposit operation using a pointer to a structure.

**Code**

#include <stdio.h>

struct Bank {

int accNo;

char name[50];

float balance;

};

void deposit(struct Bank \*b, float amount) {

b->balance += amount;

}

int main() {

struct Bank b = {123456, "Anil", 1000.0};

float amt;

printf("Enter deposit amount: ");

scanf("%f", &amt);

deposit(&b, amt);

printf("New balance for %s: %.2f\n", b.name, b.balance);

return 0;

}

**Sample Output**

Enter deposit amount: 500

New balance for Anil: 1500.00

**3. Hospital Patient Record**

**Scenario/Question**

You need to record data of multiple patients including ID, name, and disease. Display all records using an array of structures.

**Code**

#include <stdio.h>

struct Patient {

int id;

char name[30];

char disease[30];

};

int main() {

int n;

printf("Enter number of patients: ");

scanf("%d", &n);

struct Patient p[n];

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

printf("Enter ID, Name, Disease: ");

scanf("%d %s %s", &p[i].id, p[i].name, p[i].disease);

}

printf("\nPatient Records:\n");

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

printf("ID: %d, Name: %s, Disease: %s\n", p[i].id, p[i].name, p[i].disease);

}

return 0;

}

**Sample Output**

Enter number of patients: 2

Enter ID, Name, Disease: 1 Raj Fever

Enter ID, Name, Disease: 2 Rani Cold

Patient Records:

ID: 1, Name: Raj, Disease: Fever

ID: 2, Name: Rani, Disease: Cold

**✅ 4. Library System with Nested Structures**

**Scenario/Question**

Design a library system that stores book title, and author details (name, nationality) using nested structures.

**Code**

#include <stdio.h>

struct Author {

char name[30];

char nationality[20];

};

struct Book {

char title[50];

struct Author author;

};

int main() {

struct Book b;

printf("Enter title, author name, nationality: ");

scanf("%s %s %s", b.title, b.author.name, b.author.nationality);

printf("Book: %s\nAuthor: %s (%s)\n", b.title, b.author.name, b.author.nationality);

return 0;

}

**Sample Output**

Enter title, author name, nationality: CProgramming Dennis USA

Book: CProgramming

Author: Dennis (USA)

**5.** Write a program that implement structures (its operations) using i) Structures ii) Array of Structures iii) Nested Structures iv)Pointer to Structures

**C Program with All Structure Types**

#include <stdio.h>

#include <string.h>

// Simple Structure

struct Student {

int rollNo;

char name[50];

float marks;

};

// Nested Structure

struct Address {

char city[30];

int pincode;

};

struct Employee {

int id;

char name[30];

struct Address addr; // Nested structure

};

// Function using Pointer to Structure

void printStudent(struct Student \*s) {

printf("\n--- Student (Using Pointer) ---\n");

printf("Roll No: %d\n", s->rollNo);

printf("Name: %s\n", s->name);

printf("Marks: %.2f\n", s->marks);

}

int main() {

// i) Simple Structure

struct Student s1;

s1.rollNo = 101;

strcpy(s1.name, "Ravi");

s1.marks = 85.5;

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

printf("Roll No: %d\nName: %s\nMarks: %.2f\n", s1.rollNo, s1.name, s1.marks);

// ii) Array of Structures

struct Student students[2];

strcpy(students[0].name, "Anil");

students[0].rollNo = 102;

students[0].marks = 90;

strcpy(students[1].name, "Sunita");

students[1].rollNo = 103;

students[1].marks = 88;

printf("\n--- Array of Structures ---\n");

for (int i = 0; i < 2; i++) {

printf("Student %d: %s (Roll No: %d), Marks: %.2f\n",

i+1, students[i].name, students[i].rollNo, students[i].marks);

}

// iii) Nested Structures

struct Employee emp1;

emp1.id = 1;

strcpy(emp1.name, "Ajay");

strcpy(emp1.addr.city, "Hyderabad");

emp1.addr.pincode = 500081;

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

printf("ID: %d\nName: %s\nCity: %s\nPincode: %d\n",

emp1.id, emp1.name, emp1.addr.city, emp1.addr.pincode);

// iv) Pointer to Structure

struct Student \*ptr = &s1;

printStudent(ptr);

return 0;

}

**Sample Output**

--- Simple Structure ---

Roll No: 101

Name: Ravi

Marks: 85.50

--- Array of Structures ---

Student 1: Anil (Roll No: 102), Marks: 90.00

Student 2: Sunita (Roll No: 103), Marks: 88.00

--- Nested Structure ---

ID: 1

Name: Ajay

City: Hyderabad

Pincode: 500081

--- Student (Using Pointer) ---

Roll No: 101

Name: Ravi

Marks: 85.50

**6. Movie Ticket Booking System**

**Scenario/Question**

Create a structure for a movie ticket that stores movie name, seat number, and price. Display ticket details.

**Code**

#include <stdio.h>

struct Ticket {

char movie[30];

int seat;

float price;

};

int main() {

struct Ticket t = {"Inception", 15, 250.0};

printf("Movie: %s\nSeat: %d\nPrice: ₹%.2f\n", t.movie, t.seat, t.price);

return 0;

}

**Sample Output**

Movie: Inception

Seat: 15

Price: ₹250.00

**7. Bus Route Tracker**

**Scenario/Question**

Create a program that stores bus route details such as route number, start point, and destination using a structure.

**Code**

#include <stdio.h>

struct BusRoute {

int routeNo;

char start[30];

char end[30];

};

int main() {

struct BusRoute r = {101, "Hyderabad", "Bangalore"};

printf("Route No: %d\nFrom: %s\nTo: %s\n", r.routeNo, r.start, r.end);

return 0;

}

**Sample Output**

Route No: 101

From: Hyderabad

To: Bangalore

**8. Weather Monitoring System**

**Scenario/Question**

Build a structure-based weather report system to record temperature, humidity, and wind speed for a day.

**Code**

#include <stdio.h>

struct Weather {

float temperature;

int humidity;

float windSpeed;

};

int main() {

struct Weather w = {36.5, 80, 12.4};

printf("Temp: %.1f°C, Humidity: %d%%, Wind Speed: %.1f km/h\n", w.temperature, w.humidity, w.windSpeed);

return 0;

}

**Sample Output**

Temp: 36.5°C, Humidity: 80%, Wind Speed: 12.4 km/h

**9. Hotel Reservation Record**

**Scenario/Question**

Store reservation information of a hotel guest: name, room number, and number of nights. Use a structure and calculate bill assuming ₹1000 per night.

**Code**

#include <stdio.h>

struct Reservation {

char name[30];

int roomNo;

int nights;

};

int main() {

struct Reservation r = {"Sita", 202, 3};

int bill = r.nights \* 1000;

printf("Guest: %s\nRoom: %d\nBill: ₹%d\n", r.name, r.roomNo, bill);

return 0;

}

**Sample Output**

Guest: Sita

Room: 202

Bill: ₹3000

**10. Dynamic Student Entry with malloc**

**Scenario/Question**

Dynamically store student details using malloc: roll number and name, and display them.

**Code**

#include <stdio.h>

#include <stdlib.h>

struct Student {

int roll;

char name[30];

};

int main() {

struct Student \*s;

s = (struct Student\*)malloc(sizeof(struct Student));

printf("Enter roll number and name: ");

scanf("%d %s", &s->roll, s->name);

printf("Roll: %d, Name: %s\n", s->roll, s->name);

free(s);

return 0;

}

**Sample Output**

Enter roll number and name: 101 Arjun

Roll: 101, Name: Arjun