Problem 1: Dynamic Student Record Management

Objective: Manage student records using pointers to structures and dynamically allocate memory for student names.

Description:

1. Define a structure Student with fields:
    - int roll_no: Roll number
    - char *name: Pointer to dynamically allocated memory for the student's name
    - float marks: Marks obtained
2. Write a program to:
    - Dynamically allocate memory for n students.
    - Accept details of each student, dynamically allocating memory for their names.
    - Display all student details.
    - Free all allocated memory before exiting.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct Student{
   int roll_no;
   char *name;
   float marks;
};
int main()
{
   int n;
   printf("Enter number of students:");
   scanf("%d",&n);
   struct Student *students=(struct Student *)malloc(n * sizeof(struct Student));
   for(int i=0;i<n;i++)
   {
      printf("Enter the details of student %d:\n",i+1);
      printf("Enter the roll number: ");
```

```
        scanf("%d",&students[i].roll_no);
        students[i].name=(char*)malloc(100 * sizeof(char));
        printf("Enter the name:");
        scanf("%s",students[i].name);
        printf("Enter the marks:");
        scanf("%f",&students[i].marks);
    }
    printf("Display all student details:\n");
    for(int i=0;i<n;i++)
    {
        printf("Roll number:%d \n",students[i].roll_no);
        printf("Name:%s \n",students[i].name);
        printf("Marks:%f \n",students[i].marks);
    }
    for(int i=0;i<n;i++)
    {
        free(students[i].name);
    }
    free(students);
    return 0;
}
```

Output:

```
Enter number of students:2
Enter the details of student 1:
Enter the roll number: 101
Enter the name:anu
Enter the marks:79
Enter the details of student 2:
Enter the roll number: 103
Enter the name:arun
Enter the marks:80
```

Display all student details:

Roll number:101

Name:anu

Marks:79.000000

Roll number:103

Name:arun

Marks:80.000000

Problem 2: Library System with Dynamic Allocation

Objective: Manage a library system where book details are dynamically stored using pointers inside a structure.

Description:

1. Define a structure Book with fields:
    - char *title: Pointer to dynamically allocated memory for the book's title
    - char *author: Pointer to dynamically allocated memory for the author's name
    - int *copies: Pointer to the number of available copies (stored dynamically)
2. Write a program to:
    - Dynamically allocate memory for n books.
    - Accept and display book details.
    - Update the number of copies of a specific book.
    - Free all allocated memory before exiting.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct Book{
   char *title;
```

```
    char *author;
    int *copies;
};
int main()
{
    int n,book_num,new_copy;
    printf("Enter number of books:");
    scanf("%d",&n);
    getchar();
    struct Book *books=(struct Book *)malloc(n * sizeof(struct Book));
    for(int i=0;i<n;i++)
    {
        books[i].title=(char *)malloc(100 * sizeof(char));
        books[i].author=(char *)malloc(100 * sizeof(char));
        books[i].copies=(int *)malloc(100 * sizeof(int));
        printf("Enter book title %d:",i+1);
        scanf("%[^\n]",books[i].title);
        getchar();
        printf("Enter book author:");
        scanf("%[^\n]",books[i].author);
        printf("Enter the number of copies:");
        scanf("%d",books[i].copies);
        getchar();
    }
    printf("Library book details:");
    for(int i=0;i<n;i++)
    {
        printf("Book %d\n",i+1);
        printf("Title:%s\n",books[i].title);
        printf("Author:%s\n",books[i].author);
        printf("Copies:%d \n",*books[i].copies);
```

```
    }
    printf("Enter book number to update copies:\n");
    scanf("%d",&book_num);
    if(book_num>=1 && book_num<=n)
    {
        printf("Enter the number of new copies for book %d:",book_num);
        scanf("%d",&new_copy);
        *books[book_num-1].copies=new_copy;
        printf("Updated number of copies for book %d: %d\n",book_num,*books[book_num-1].copies);
    }
    else
    {
        printf("Invalid book number");
    }
    for(int i=0;i<n;i++)
    {
        free(books[i].title);
        free(books[i].author);
        free(books[i].copies);
    }
    free(books);
    return 0;
}
```

Output:

```
Enter number of books:2
Enter book title 1:Harry potter
Enter book author:Jk rowling
Enter the number of copies:5
Enter book title 2:Fault in our stars
Enter book author:John green
Enter the number of copies:4
```

Library book details:Book 1

Title:Harry potter

Author:Jk rowling

Copies:5

Book 2

Title:Fault in our stars

Author:John green

Copies:4

Enter book number to update copies:

2

Enter the number of new copies for book 2:9

Updated number of copies for book 2: 9

Problem 1: Complex Number Operations

Objective: Perform addition and multiplication of two complex numbers using structures passed to functions.

Description:

1. Define a structure Complex with fields:
    - float real: Real part of the complex number
    - float imag: Imaginary part of the complex number
2. Write functions to:
    - Add two complex numbers and return the result.
    - Multiply two complex numbers and return the result.
3. Pass the structures as arguments to these functions and display the results.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct Complex{
   float real;
```

```
    float imag;
};
struct Complex add(struct Complex c1, struct Complex c2);
struct Complex multiply(struct Complex c1,struct Complex c2);
int main()
{
    struct Complex c1={5,9}, c2={4,3};
    struct Complex sum=add(c1,c2);
    struct Complex product=multiply(c1,c2);
    printf("sum:%.2f+%.2f \n",sum.real,sum.imag);
    printf("product:%.2f*%.2f",product.real,product.imag);
    return 0;
}
struct Complex add(struct Complex c1, struct Complex c2)
{
    struct Complex res;
    res.real=c1.real+c2.real;
    res.imag=c1.imag+c2.imag;
    return res;
}
struct Complex multiply(struct Complex c1,struct Complex c2)
{
    struct Complex res;
    res.real=c1.real*c2.real - c1.imag*c2.imag;
    res.imag=c1.real*c2.real + c1.imag*c2.imag;
    return res;
}
```

Output:

sum:9.00+12.00

product:-7.00*47.00

Problem 2: Rectangle Area and Perimeter Calculator

Objective: Calculate the area and perimeter of a rectangle by passing a structure to functions.

Description:

1. Define a structure Rectangle with fields:
    - float length: Length of the rectangle
    - float width: Width of the rectangle
2. Write functions to:
    - Calculate and return the area of the rectangle.
    - Calculate and return the perimeter of the rectangle.
3. Pass the structure to these functions by value and display the results in main.

```
#include<stdio.h>
#include<stdlib.h>
struct Rectangle{
   float length;
   float width;
};
float area(struct Rectangle rect);
float perimeter(struct Rectangle rect);
int main()
{
   struct Rectangle rect={6,4.5};
   printf("Area of rectangle:%.2f\n",area(rect));
   printf("Perimeter of rectangle:%.2f\n",perimeter(rect));
}
float area(struct Rectangle rect)
{
   return rect.length*rect.width;
}
float perimeter(struct Rectangle rect)
{
```

```
    return 2*(rect.length+rect.width);
}
```

Output:

Area of rectangle:27.00

Perimeter of rectangle:21.00

Problem 3: Student Grade Calculation

Objective: Calculate and assign grades to students based on their marks by passing a structure to a function.

Description:

1. Define a structure Student with fields:
    - char name[50]: Name of the student
    - int roll_no: Roll number
    - float marks[5]: Marks in 5 subjects
    - char grade: Grade assigned to the student
2. Write a function to:
    - Calculate the average marks and assign a grade (A, B, etc.) based on predefined criteria.
3. Pass the structure by reference to the function and modify the grade field.

```
 #include<stdio.h>
#include<stdlib.h>
struct Student{
    char name[50];
    int roll_no;
    float marks[5];
    char grade;
};
char calculateGrade(struct Student* student);
int main()
{
```

```
    struct Student student={"Arun",101,{70,75,80,85,90},' '};
    calculateGrade(&student);
    printf("Student name:%s\n",student.name);
    printf("Roll number:%d\n",student.roll_no);
    printf("Grade:%c\n",student.grade);
    return 0;
}
char calculateGrade(struct Student* student)
{
    float total=0;
    for(int i=0;i<5;i++)
    {
        total+=student->marks[i];
    }
    float average=total/5;
    if(average>=90)
    {
        student->grade ='A';
    }
    else if(average>=75)
    {
        student->grade='B';
    }
    else if(average>=60)
    {
        student->grade='C';
    }
    else if(average>=50)
    {
        student->grade='D';
    }
```

```
    else
    {
        student->grade='F';
    }
}
```

Output:

Student name:Arun

Roll number:101

Grade:B

Problem 4: Point Operations in 2D Space

Objective: Calculate the distance between two points and check if a point lies within a circle using structures.

Description:

1. Define a structure Point with fields:
    - float x: X-coordinate of the point
    - float y: Y-coordinate of the point
2. Write functions to:
    - Calculate the distance between two points.
    - Check if a given point lies inside a circle of a specified radius (center at origin).
3. Pass the Point structure to these functions and display the results.

```
 #include<stdio.h>
#include<math.h>
struct PointOperation{
    float x;
    float y;
};
float calculateDistance(struct PointOperation p1, struct PointOperation p2);
int InsideCircle(struct PointOperation p,float radius);
```

```
int main()
{
   float dis,radius=3;
   struct PointOperation p1={5,8};
   struct PointOperation p2={8,9};
   dis=calculateDistance(p1,p2);
   printf("Distance between points(%.2f,%.2f) and (%.2f,%.2f) is:%.2f\n",p1.x,p1.y,p2.x,p2.y,dis);
   if(InsideCircle(p1,radius))
   {
      printf("Point(%.2f,%.2f) is inside the circle.\n",p1.x,p1.y);
   }
   else
   {
      printf("Point(%.2f,%.2f) is outside the circle.\n",p1.x,p1.y);
   }
   if(InsideCircle(p2,radius))
   {
      printf("Point(%.2f,%.2f) is inside the circle.\n",p2.x,p2.y);
   }
   else
   {
      printf("Point(%.2f,%.2f) is outside the circle.\n",p2.x,p2.y);
   }
   return 0;
}

float calculateDistance(struct PointOperation p1, struct PointOperation p2)
{
   return sqrt((p2.x-p1.x)*(p2.x-p1.x)+(p2.y-p1.y)*(p2.y-p1.y));
}
int InsideCircle(struct PointOperation p,float radius)
```

```
{
   float dis=sqrt(p.x*p.x+p.y*p.y);
   return dis<=radius;
}
```

Output:

Distance between points(5.00,8.00) and (8.00,9.00) is:3.16

Point(5.00,8.00) is outside the circle.

Point(8.00,9.00) is outside the circle.

Problem 5: Employee Tax Calculation

Objective: Calculate income tax for an employee based on their salary by passing a structure to a function.

Description:

1. Define a structure Employee with fields:
    - char name[50]: Employee name
    - int emp_id: Employee ID
    - float salary: Employee salary
    - float tax: Tax to be calculated (initialized to 0)
2. Write a function to:
    - Calculate tax based on salary slabs (e.g., 10% for salaries below $50,000, 20% otherwise).
    - Modify the tax field of the structure.
3. Pass the structure by reference to the function and display the updated tax in main.

has context menu

```
#include<stdio.h>
#include<stdlib.h>
struct Employee
{
   char name[50];
```

```
    int emp_id;
    float salary;
    float tax;
};
float calculateTax(struct Employee* emp);
int main()
{
    struct Employee emp={"Arun",110,60000,0};
    calculateTax(&emp);
    printf("Employee name:%s\n",emp.name);
    printf("Employee Id:%d\n",emp.emp_id);
    printf("Employee salary:%.2f\n",emp.salary);
    printf("Tax calculated:%.2f\n",emp.tax);
    return 0;
}
float calculateTax(struct Employee* emp)
{
    if(emp->salary<50000)
    {
        emp->tax=emp->salary*0.10;
    }
    else
    {
        emp->tax=emp->salary*0.20;
    }
}
```

Output:

```
Employee name:Arun
Employee Id:110
Employee salary:60000.00
Tax calculated:12000.00
```

Problem Statement: Vehicle Service Center Management Objective: Build a system to manage vehicle servicing records using nested structures. Description: Define a structure Vehicle with fields: char license_plate[15]: Vehicle's license plate number char owner_name[50]: Owner's name char vehicle_type[20]: Type of vehicle (e.g., car, bike) Define a nested structure Service inside Vehicle with fields: char service_type[30]: Type of service performed float cost: Cost of the service char service_date[12]: Date of service Implement the following features: Add a vehicle to the service center record. Update the service history for a vehicle. Display the service details of a specific vehicle. Generate and display a summary report of all vehicles serviced, including total revenue.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct Service{
   char service_type[30];
   float cost;
   char service_date[12];
   char license_plate[15];
};
struct Vehicle{
   char license_plate[15];
   char owner_name[50];
   char vehicle_type[20];
};
struct Vehicle vehicles[100];
struct Service services[1000];
int vehicle_count=0;
int service_count=0;
int main()
{
   int choice=0;
   while(choice!=5)
   {
      printf("Vehicle Service centre management \n");
      printf("1.Add a vehicle \n");
```

```
printf("2.Add services to a vehicle \n");
printf("3.Display service details of a vehicle \n");
printf("4.Display summary report \n");
printf("5.Exit \n");
printf("Enter your choice:");
scanf("%d",&choice);
if(choice==1)
{
   printf("Enter license plate:");
   scanf("%s",vehicles[vehicle_count].license_plate);
   getchar();
   printf("Enter owner's name:");
   scanf("%[^\n]",vehicles[vehicle_count].owner_name);
   printf("Enter vehicle type(car/bike):");
   scanf("%s",vehicles[vehicle_count].vehicle_type);
   vehicle_count++;
   printf("Vehicle added successfully \n");
}
else if(choice==2)
{
   char license[15];
   printf("Enter the license plate of vehicle:");
   scanf("%s",license);
   int found=0,i=0;
   while(i<vehicle_count)
   {
      if(strcmp(vehicles[i].license_plate,license)==0){
         found = 1;
         break;
      }
      i++;
```

```
    }
    if(!found)
    {
        printf("Vehicle not found");
        continue;
    }
    getchar();
    printf("Enter service type:");
    scanf("%[^\n]",services[service_count].service_type);
    printf("Enter service cost:");
    scanf("%f",&services[service_count].cost);
    printf("Enter service date(dd-mm--yyyy):");
    scanf("%s",services[service_count].service_date);
    strcpy(services[service_count].license_plate,license);
    service_count++;
    printf("Services added successfully \n");
}
else if(choice==3)
{
    char license[15];
    printf("Enter the license plate of vehicle:");
    scanf("%s",license);
    printf("Service details of vehicles %s:\n",license);
    int found=0,i=0;
    while(i<service_count)
    {
        if(strcmp(services[i].license_plate,license)==0)
        {
            printf("Service type:%s \n",services[i].service_type);
            printf("Cost:%.2f \n",services[i].cost);
            printf("Date:%s \n",services[i].service_date);
```

```
            found=1;
        }
        i++;
    }
    if(!found)
    {
        printf("Services not found");
    }
}
else if(choice==4)
{
    printf("Summary report \n");
    float total_revenue=0;
    int i=0;
    while(i<vehicle_count)
    {
        printf("vehicle %d:\n",i+1);
        printf("License plate:%s \n",vehicles[i].license_plate);
        printf("Owner's name:%s\n",vehicles[i].owner_name);
        printf("Vehicle type:%s\n",vehicles[i].vehicle_type);
        int service_found=0,j=0;
        while(j<service_count)
        {
            if(strcmp(services[j].license_plate,vehicles[i].license_plate)==0)
            {
                printf("Service type:%s, Cost:%.2f, Date:%s
\n",services[j].service_type,services[j].cost,services[j].service_date);
                total_revenue+=services[j].cost;
                service_found=1;
            }
            j++;
```

```
            }
            if(!service_found)
            {
                printf("No services found \n");
            }
            i++;
        }
        printf("Total revenue:%.2f\n",total_revenue);
    }
    else if(choice==5)
    {
        printf("Exit \n");
    }
    else
    {
        printf("Invalid choice");
    }
  }
  return 0;
}
```

Output:

```
Vehicle Service centre management
1.Add a vehicle
2.Add services to a vehicle
3.Display service details of a vehicle
4.Display summary report
5.Exit
Enter your choice:1
Enter license plate:kl24A7895
Enter owner's name:Anu
Enter vehicle type(car/bike):Car
```

Vehicle added successfully

Vehicle Service centre management

1.Add a vehicle

2.Add services to a vehicle

3.Display service details of a vehicle

4.Display summary report

5.Exit

Enter your choice:1

Enter license plate:Kl52B9604

Enter owner's name:Arun

Enter vehicle type(car/bike):Bike

Vehicle added successfully

Vehicle Service centre management

1.Add a vehicle

2.Add services to a vehicle

3.Display service details of a vehicle

4.Display summary report

5.Exit

Enter your choice:2

Enter the license plate of vehicle:Kl52B9604

Enter service type:Colour change

Enter service cost:40000

Enter service date(dd-mm--yyyy):23-05-2024

Services added successfully

Vehicle Service centre management

1.Add a vehicle

2.Add services to a vehicle

3.Display service details of a vehicle

4.Display summary report

5.Exit

Enter your choice:2

Enter the license plate of vehicle:kl24A7895

Enter service type:Oil change

Enter service cost:60000

Enter service date(dd-mm--yyyy):17-07-2024

Services added successfully

Vehicle Service centre management

1.Add a vehicle

2.Add services to a vehicle

3.Display service details of a vehicle

4.Display summary report

5.Exit

Enter your choice:3

Enter the license plate of vehicle:kl24A7895

Service details of vehicles kl24A7895:

Service type:Oil change

Cost:60000.00

Date:17-07-2024

Vehicle Service centre management

1.Add a vehicle

2.Add services to a vehicle

3.Display service details of a vehicle

4.Display summary report

5.Exit

Enter your choice:3

Enter the license plate of vehicle:Kl52B9604

Service details of vehicles Kl52B9604:

Service type:Colour change

Cost:40000.00

Date:23-05-2024

Vehicle Service centre management

1.Add a vehicle

2.Add services to a vehicle

3.Display service details of a vehicle

4.Display summary report

5.Exit

Enter your choice:4

Summary report

vehicle 1:

License plate:kl24A7895

Owner's name:Anu

Vehicle type:Car

Service type:Oil change, Cost:60000.00, Date:17-07-2024

vehicle 2:

License plate:Kl52B9604

Owner's name:Arun

Vehicle type:Bike

Service type:Colour change, Cost:40000.00, Date:23-05-2024

Total revenue:100000.00

Vehicle Service centre management

1.Add a vehicle

2.Add services to a vehicle

3.Display service details of a vehicle

4.Display summary report

5.Exit

Enter your choice:5

Exit