DAY 12-DAILY ASSIGNMENTS

ANSU MARIUM SHIBU

21-11-2024

1. Problem 1: Dynamic Array Resizing

Objective: Write a program to dynamically allocate an integer array and allow the user to resize it.

Description:

The program should ask the user to enter the initial size of the array.

Allocate memory using malloc.

Allow the user to enter elements into the array.

Provide an option to increase or decrease the size of the array. Use realloc to adjust the size.

Print the elements of the array after each resizing operation.

```c
#include<stdio.h>
#include<stdlib.h>

int main() {
    int *arr;
    int size, new_size, i, choice;

    printf("Enter initial size of array: ");
    scanf("%d", &size);
    arr = (int *)malloc(size * sizeof(int));

    if (arr == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    printf("Enter %d elements:\n", size);
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Array before resizing:\n");
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
```

```c
    printf("\n");

    do {
        printf("Choose an option:\n");
        printf("1. Increase the size\n");
        printf("2. Decrease the size\n");
        printf("3. Exit\n");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter new size of array: ");
                scanf("%d", &new_size);

                arr = (int *)realloc(arr, new_size * sizeof(int));

                if (arr == NULL) {
                    printf("Memory allocation failed!\n");
                    return 1;
                }

                if (new_size > size) {
                    printf("Enter %d new elements:\n", new_size - size);
                    for (i = size; i < new_size; i++) {
```

```c
                if (new_size > size) {
                    printf("Enter %d new elements:\n", new_size - size);
                    for (i = size; i < new_size; i++) {
                        printf("Enter element %d: ", i + 1);
                        scanf("%d", &arr[i]);
                    }
                }

                printf("Array after resizing (increased):\n");
                for (i = 0; i < new_size; i++) {
                    printf("%d ", arr[i]);
                }
                printf("\n");
                size = new_size;
                break;

            case 2:
                printf("Enter new size of array: ");
                scanf("%d", &new_size);

                arr = (int *)realloc(arr, new_size * sizeof(int));
```

```c
            case 2:
                printf("Enter new size of array: ");
                scanf("%d", &new_size);

                arr = (int *)realloc(arr, new_size * sizeof(int));

                if (arr == NULL) {
                    printf("Memory allocation failed!\n");
                    return 1;
                }

                printf("Array after resizing (decreased):\n");
                for (i = 0; i < new_size; i++) {
                    printf("%d ", arr[i]);
                }
                printf("\n");
                size = new_size;
                break;

            case 3:
                printf("Exiting the program\n");
```

```c
76                  printf("%d ", arr[i]);
77                }
78                printf("\n");
79                size = new_size;
80                break;
81
82            case 3:
83                printf("Exiting the program\n");
84                break;
85
86            default:
87                printf("Invalid option!\n");
88        }
89    } while (choice != 3);
90
91    free(arr);
92    return 0;
93 }
94
```

```
PS D:\c progrms coding> gcc dynamicass1.c
PS D:\c progrms coding> ./a
Enter initial size of array: 5
Enter 5 elements:
1
2
3
4
5
Array before resizing:
1 2 3 4 5
Choose an option:
1. Increase the size
2. Decrease the size
3. Exit
1
Enter new size of array: 8
Enter 3 new elements:
Enter element 6: 6
Enter element 7: 8
Enter element 8: 10
Array after resizing (increased):
1 2 3 4 5 6 8 10
Choose an option:
1. Increase the size
2. Decrease the size
3. Exit
2
```

```
Choose an option:
1. Increase the size
2. Decrease the size
3. Exit
2
Enter new size of array: 4
Array after resizing (decreased):
1 2 3 4
Choose an option:
1. Increase the size
Enter new size of array: 4
Array after resizing (decreased):
1 2 3 4
Choose an option:
1. Increase the size
1 2 3 4
Choose an option:
1. Increase the size
Choose an option:
1. Increase the size
1. Increase the size
2. Decrease the size
2. Decrease the size
3. Exit
3
Exiting the program
PS D:\c progrms coding> []
```

2.Problem 2: String Concatenation Using Dynamic Memory

Objective: Create a program that concatenates two strings using dynamic memory allocation.

Description:

Accept two strings from the user.

Use malloc to allocate memory for the first string.

Use realloc to resize the memory to accommodate the concatenated string.

Concatenate the strings and print the result.

Free the allocated memory.

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

int main(){
    char *str1,*str2,*res;
    int len1,len2;

    printf("enetr the frst string:\n");
    str1=(char*)malloc(100 *sizeof(char));

    if(str1==NULL){
     printf("memory allocate fail\n");
     return 1;
    }
    scanf("%s",str1);

    printf("enter sec str:\n");
    str2=(char*)malloc(100 *sizeof(char));

    if(str2==NULL){
     printf("memory allocate fail\n");
     return 1;
    }
    scanf("%s",str2);

    len1=strlen(str1);
    len2=strlen(str2);
```

```c
    len1=strlen(str1);
    len2=strlen(str2);

    res = (char *)malloc((len1 + len2 + 1) * sizeof(char));
    if(res==NULL){
    printf("memory allocate fail\n");
    free(str1);
    free(str2);
    return 1;
    }
    strcpy(res,str1);
    strcat(res,str2);
    printf("Concatenated string: %s\n", res);

    free(str1);
    free(str2);
    free(res);

    return 0;
}
```

```
PS D:\c progrms coding> ./a
enetr the frst string:
ansu
enter sec str:
shibu
Concatenated string: ansushibu
```

3. Problem 5: Dynamic 2D Array Allocation

Objective: Write a program to dynamically allocate a 2D array.

Description:

Accept the number of rows and columns from the user.

Use malloc (or calloc) to allocate memory for the rows and columns dynamically.

Allow the user to input values into the 2D array.

Print the array in matrix format.

Free all allocated memory at the end.

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int **arr;
    int rows, cols, i, j;

    printf("Enter number of rows: ");
    scanf("%d", &rows);
    printf("Enter number of columns: ");
    scanf("%d", &cols);

    arr = (int **)malloc(rows * sizeof(int *));
    for (i = 0; i < rows; i++) {
        arr[i] = (int *)malloc(cols * sizeof(int));
    }

    printf("Enter elements:\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            scanf("%d", &arr[i][j]);
        }
    }

    printf("The matrix is:\n");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            printf("%d ", arr[i][j]);
```

```c
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }

    for (i = 0; i < rows; i++) {
        free(arr[i]);
    }
    free(arr);

    return 0;
```

```
PS D:\c progrms coding> gcc dynamicassi5.c
PS D:\c progrms coding> ./a
Enter number of rows: 3
Enter number of columns: 3
Enter elements:
1 2 3
4 5 6
7 8 9
The matrix is:
1 2 3
4 5 6
7 8 9
PS D:\c progrms coding> gcc dynamicassi5.c
PS D:\c progrms coding> ./a
Enter number of rows: 3
Enter number of columns: 3
Enter elements:
```

.

4 Problem 3: Sparse Matrix Representation

Objective: Represent a sparse matrix using dynamic memory allocation.

Description:

Accept a matrix of size m×nm \times nm×n from the user.

Store only the non-zero elements in a dynamically allocated array of structures (with fields for row, column, and value).

Print the sparse matrix representation.

Free the allocated memory at the end.

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int m, n;
    printf("Enter the number of rows: ");
    scanf("%d", &m);
    printf("Enter the number of columns: ");
    scanf("%d", &n);

    int matrix[m][n];
    printf("Enter the elements of the matrix:\n");

    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    int *sparseMatrix = (int*)malloc(m * n * sizeof(int));
    int count = 0;

    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            if (matrix[i][j] != 0) {
                sparseMatrix[count++] = i;
                sparseMatrix[count++] = j;
                sparseMatrix[count++] = matrix[i][j];
            }
        }
    }

    printf("\nSparse Matrix Representation:\n");
    printf("Row\tColumn\tValue\n");

    for (int i = 0; i < count; i += 3) {
        printf("%d\t%d\t%d\n", sparseMatrix[i], sparseMatrix[i + 1], sparseMatrix[i + 2]);
    }

    free(sparseMatrix);

    return 0;
}
```

```
PS D:\c progrms coding> gcc dynamicassi3.c
PS D:\c progrms coding> ./a
Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the matrix:
1 0 0
0 0 5
0 6 0

Sparse Matrix Representation:
Row      Column  Value
0        0        1
1        2        5
2        1        6
PS D:\c progrms coding>
```

5 Problem 1: Student Record Management System Objective Create a program to manage student records using structures. Requirements 1. Define a Student structure with the following fields: char name[50] int rollNumber float marks 2. Implement functions to: Add a new student record. Display all student records. Find and display a student record by roll number. Calculate and display the average marks of all students. 3. Implement a menu-driven interface to perform the above operations. use struct

```c
#include <stdio.h>
#include <string.h>

struct Student {
    char name[50];
    int rollNumber;
    float marks;
};

int main() {
    struct Student students[100];
    int count = 0;
    int choice;

    do {
        printf("\nMenu:\n");
        printf("1. Add Student\n");
        printf("2. Display All Students\n");
        printf("3. Find Student by Roll Number\n");
        printf("4. Calculate Average Marks\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
```

```c
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
            printf("Enter name: ");
            scanf(" %[^\n]", students[count].name);
            printf("Enter roll number: ");
            scanf("%d", &students[count].rollNumber);
            printf("Enter marks: ");
            scanf("%f", &students[count].marks);
            count++;
        } else if (choice == 2) {
            for (int i = 0; i < count; i++) {
                printf("Name: %s, Roll Number: %d, Marks: %.2f\n",
                    students[i].name, students[i].rollNumber, students[i].marks);
            }
        } else if (choice == 3) {
            int roll;
            printf("Enter roll number to find: ");
            scanf("%d", &roll);
            for (int i = 0; i < count; i++) {
                if (students[i].rollNumber == roll) {
                    printf("Name: %s, Roll Number: %d, Marks: %.2f\n"
```

```c
        }
    } else if (choice == 3) {
        int roll;
        printf("Enter roll number to find: ");
        scanf("%d", &roll);
        for (int i = 0; i < count; i++) {
            if (students[i].rollNumber == roll) {
                printf("Name: %s, Roll Number: %d, Marks: %.2f\n",
                       students[i].name, students[i].rollNumber, students[i].marks);
                break;
            }
        }
    } else if (choice == 4) {
        float total = 0;
        for (int i = 0; i < count; i++) {
            total += students[i].marks;
        }
        printf("Average Marks: %.2f\n", total / count);
    }
} while (choice != 5);
```

```c
        printf("Average Marks: %.2f\n", total / count);
    }
} while (choice != 5);

    return 0;
}
```

```
Enter your choice: 5
PS D:\c progrms coding> gcc struct2.c
PS D:\c progrms coding> ./a

Menu:
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 1
Enter name: ansu
Enter roll number: 23
Enter marks: 31

Menu:
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
```

```
Menu:
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 1
Enter name: shibu
Enter roll number: 32
Enter marks: 45

Menu:
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 2
Name: ansu, Roll Number: 23, Marks: 31.00
```

```
Menu:
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 2
Name: ansu, Roll Number: 23, Marks: 31.00
Name: shibu, Roll Number: 32, Marks: 45.00

Menu:
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 3
Enter roll number to find: 32
Name: shibu, Roll Number: 32, Marks: 45.00
```

```
Menu:
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 4
Average Marks: 38.00

Menu:
```

6. Problem 1: Employee Management System

Objective: Create a program to manage employee details using structures.

Description:

Define a structure Employee with fields:

int emp_id: Employee ID

char name[50]: Employee name

float salary: Employee salary

Write a menu-driven program to:

Add an employee.

Update employee salary by ID.

Display all employee details.

Find and display details of the employee with the highest salary.

```c
#include <stdio.h>

struct Employee {
    int emp_id;
    char name[50];
    float salary;
};

int main() {
    struct Employee employees[100];
    int count = 0, choice, id, i;
    float max_salary;
    int max_index;

    do {
        printf("\nMenu:\n");
        printf("1. Add Employee\n");
        printf("2. Update Salary by ID\n");
        printf("3. Display All Employees\n");
        printf("4. Find Employee with Highest Salary\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
            printf("Enter Employee ID: ");
            scanf("%d", &employees[count].emp_id);
            printf("Enter Name: ");
            scanf(" %[^\n]", employees[count].name);
```

```c
    if (choice == 1) {
        printf("Enter Employee ID: ");
        scanf("%d", &employees[count].emp_id);
        printf("Enter Name: ");
        scanf(" %[^\n]", employees[count].name);
        printf("Enter Salary: ");
        scanf("%f", &employees[count].salary);
        count++;
    }
    else if (choice == 2) {
        printf("Enter Employee ID to update salary: ");
        scanf("%d", &id);
        for (i = 0; i < count; i++) {
            if (employees[i].emp_id == id) {
                printf("Enter new salary: ");
                scanf("%f", &employees[i].salary);
                break;
            }
        }
    }
    else if (choice == 3) {
        for (i = 0; i < count; i++) {
            printf("ID: %d, Name: %s, Salary: %.2f\n",
                employees[i].emp_id, employees[i].name, employees[i].salary);
        }
    }
```

```c
    else if (choice == 3) {
        for (i = 0; i < count; i++) {
            printf("ID: %d, Name: %s, Salary: %.2f\n",
                employees[i].emp_id, employees[i].name, employees[i].salary);
        }
    }
    else if (choice == 4) {
        max_salary = employees[0].salary;
        max_index = 0;
        for (i = 1; i < count; i++) {
            if (employees[i].salary > max_salary) {
                max_salary = employees[i].salary;
                max_index = i;
            }
        }
        printf("Employee with Highest Salary: ID: %d, Name: %s, Salary: %.2f\n",
            employees[max_index].emp_id, employees[max_index].name, employees[max_index].salary);
    }

} while (choice != 5);

return 0;
}
```

```
PS D:\c progrms coding> gcc structmenuassi1.c
PS D:\c progrms coding> ./a

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice: 1
Enter Employee ID: 123
Enter Name: ansu
Enter Salary: 20000

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
```

```
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice: 1
Enter Employee ID: 145
Enter Name: shibu
Enter Salary: 25000

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice: 3
ID: 123, Name: ansu, Salary: 20000.00
ID: 145, Name: shibu, Salary: 25000.00
```

```
ID: 123, Name: ansu, Salary: 20000.00
ID: 145, Name: shibu, Salary: 25000.00

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice: 2
Enter Employee ID to update salary: 123
Enter new salary: 30000

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice:
```

```
Enter your choice: 2
Enter Employee ID to update salary: 123
Enter new salary: 30000

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice: 4
Employee with Highest Salary: ID: 123, Name: ansu, Salary: 30000.00

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice:
```

```
Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice: 4
Employee with Highest Salary: ID: 123, Name: ansu, Salary: 30000.00

Menu:
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Find Employee with Highest Salary
5. Exit
Enter your choice: 5
PS D:\c progrms coding>
```

7. Problem 2: Library Management System

Objective: Manage a library system with a structure to store book details.

Description:

Define a structure Book with fields:

int book_id: Book ID

char title[100]: Book title

char author[50]: Author name

int copies: Number of available copies

Write a program to:

Add books to the library.

Issue a book by reducing the number of copies.

Return a book by increasing the number of copies.

Search for a book by title or author name.

```c
#include <stdio.h>
#include <string.h>

struct Book {
    int book_id;
    char title[100];
    char author[50];
    int copies;
};

int main() {
    struct Book library[100];
    int count = 0, choice;

    do {
        printf("\n1. Add Book\n2. Issue Book\n3. Return Book\n4. Search Book\n5. Exit\nEnter choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
            printf("Enter Book ID: ");
```

```c
        else if (choice == 2) {
            for (int i = 0; i < count; i++) {
                if (library[i].book_id == id && library[i].copies > 0) {
                    break;
                }
            }
        }
        else if (choice == 3) {
            int id;
            printf("Enter Book ID to return: ");
            scanf("%d", &id);
            for (int i = 0; i < count; i++) {
                if (library[i].book_id == id) {
                    library[i].copies++;
                    printf("Book returned.\n");
                    break;
                }
            }
        }
        else if (choice == 4) {
            char query[100];
            printf("Enter Title or Author: ");
            scanf(" %[^\n]", query);
```

```c
            }
        }
        else if (choice == 4) {
            char query[100];
            printf("Enter Title or Author: ");
            scanf(" %[^\n]", query);
            for (int i = 0; i < count; i++) {
                if (strstr(library[i].title, query) || strstr(library[i].author, query)) {
                    printf("ID: %d, Title: %s, Author: %s, Copies: %d\n",
                        library[i].book_id, library[i].title, library[i].author, library[i].copies);
                }
            }
        }
    } while (choice != 5);

    return 0;
}
```

```
5. Exit
Enter your choice: 5
PS D:\c progrms coding> gcc structmenuassi2.c
PS D:\c progrms coding> ./a

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Exit
Enter choice: 1
Enter Book ID: 123
Enter Title: andrea
Enter Author: sree
Enter Copies: 2
```

```
1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Exit
Enter choice: 1
Enter Book ID: 145
Enter Title: swaga
Enter Author: hari
Enter Copies: 5

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Exit
Enter choice:
```

```
Enter Title: swaga
Enter Author: hari
Enter Copies: 5

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Exit
Enter choice: 2
Enter Book ID to issue: 123
Book issued.

1. Add Book
2. Issue Book
```

```
1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Exit
Enter choice: 3
Enter Book ID to return: 145
Book returned.
```

```
1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Exit
Enter choice: 4
Enter Title or Author: hari
ID: 145, Title: swaga, Author: hari, Copies: 6

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
```

8.Problem 3: Cricket Player Statistics

Objective: Store and analyze cricket player performance data.

Description:

Define a structure Player with fields:

char name[50]: Player name

int matches: Number of matches played

int runs: Total runs scored

float average: Batting average

Write a program to:

Input details for n players.

Calculate and display the batting average for each player.

Find and display the player with the highest batting average.

```c
#include <stdio.h>

struct Player {
    char name[50];
    int matches;
    int runs;
    float average;
};

int main() {
    int n;
    printf("Enter the number of players: ");
    scanf("%d", &n);

    struct Player players[n];
    int highestIndex = 0;

    for (int i = 0; i < n; i++) {
        printf("Enter details for player %d\n", i + 1);
        printf("Name: ");
        scanf(" %[^\n]", players[i].name);
        printf("Matches played: ");
        scanf("%d", &players[i].matches);
        printf("Total runs: ");
        scanf("%d", &players[i].runs);
```

```c
    for (int i = 0; i < n; i++) {
        printf("Total runs: ");
        scanf("%d", &players[i].runs);

        if (players[i].matches > 0)
            players[i].average = (float)players[i].runs / players[i].matches;
        else
            players[i].average = 0;

        if (players[i].average > players[highestIndex].average) {
            highestIndex = i;
        }
    }

    printf("\nPlayer Statistics:\n");
    for (int i = 0; i < n; i++) {
        printf("Name: %s, Matches: %d, Runs: %d, Average: %.2f\n",
            players[i].name, players[i].matches, players[i].runs, players[i].average);
    }

    printf("\nPlayer with the highest batting average:\n");
    printf("Name: %s, Average: %.2f\n", players[highestIndex].name, players[highestIndex].average);

    return 0;
```

```
PS D:\c progrms coding> gcc structasss3.c
PS D:\c progrms coding> ./a
Enter the number of players: 4
Enter details for player 1
Name: ansu
Matches played: 5
Total runs: 23
Enter details for player 2
Name: shibu
Matches played: 6
Total runs: 23
Enter details for player 3
Name: marium
Matches played: 7
Total runs: 3
Enter details for player 4
Name: daan
Matches played: 9
Total runs: 23

Player Statistics:
Name: ansu, Matches: 5, Runs: 23, Average: 4.60
Name: shibu, Matches: 6, Runs: 23, Average: 3.83
Name: marium, Matches: 7, Runs: 3, Average: 0.43
Name: daan, Matches: 9, Runs: 23, Average: 2.56

Player with the highest batting average:
Name: ansu, Average: 4.60
PS D:\c progrms coding>
```

.

9.Problem 4: Student Grading System

Objective: Manage student data and calculate grades based on marks.

Description:

Define a structure Student with fields:

int roll_no: Roll number

char name[50]: Student name

float marks[5]: Marks in 5 subjects

char grade: Grade based on the average marks

Write a program to:

Input details of n students.

Calculate the average marks and assign grades (A, B, C, etc.).

Display details of students along with their grades.

```c
#include <stdio.h>

struct Student {
    int roll_no;
    char name[50];
    float marks[5];
    char grade;
};

int main() {
    int n;
    printf("Enter the number of students: ");
    scanf("%d", &n);

    struct Student students[n];

    for (int i = 0; i < n; i++) {                    (char [44])"Enter roll number and name for student %d: "
        printf("Enter roll number and name for student %d: ", i + 1);
        scanf("%d %[^\n]", &students[i].roll_no, students[i].name);

        float total = 0;
        printf("Enter marks for 5 subjects: ");
        for (int j = 0; j < 5; j++) {
            scanf("%f", &students[i].marks[j]);
            total += students[i].marks[j];
        }

        float average = total / 5;
        if (average >= 90) students[i].grade = 'A';
```

```c
    for (int i = 0; i < n; i++) {
        scanf("%d %[^\n]", &students[i].roll_no, students[i].name);

        float total = 0;
        printf("Enter marks for 5 subjects: ");
        for (int j = 0; j < 5; j++) {
            scanf("%f", &students[i].marks[j]);
            total += students[i].marks[j];
        }

        float average = total / 5;
        if (average >= 90) students[i].grade = 'A';
        else if (average >= 75) students[i].grade = 'B';
        else if (average >= 50) students[i].grade = 'C';
        else students[i].grade = 'D';
    }

    printf("\nStudent Details:\n");
    for (int i = 0; i < n; i++) {
        printf("Roll No: %d, Name: %s, Grade: %c\n", students[i].roll_no, students[i].name, students[i].grade);
    }

    return 0;
}
```

```
PS D:\c progrms coding> gcc structasss4.c
PS D:\c progrms coding> ./a
Enter the number of students: 2
Enter roll number and name for student 1: 23
ansu
Enter marks for 5 subjects: 23
41
5
67
100
Enter roll number and name for student 2: 45
thara
Enter marks for 5 subjects: 45
100
91
95
81

Student Details:
Roll No: 23, Name: ansu, Grade: D
Roll No: 45, Name: thara, Grade: B
PS D:\c progrms coding>
```

10. Problem 5: Flight Reservation System

Objective: Simulate a simple flight reservation system using structures.

Description:

Define a structure Flight with fields:

char flight_number[10]: Flight number

char destination[50]: Destination city

int available_seats: Number of available seats

Write a program to:

Add flights to the system.

Book tickets for a flight, reducing available seats accordingly.

Display the flight details based on destination.

Cancel tickets, increasing the number of available seats

```c
#include <stdio.h>
#include <string.h>

struct Flight {
    char flight_number[10];
    char destination[50];
    int available_seats;
};

int main() {
    struct Flight flights[5];
    int flight_count = 0;
    int choice;

    do {
        printf("\nFlight Reservation System Menu:\n");
        printf("1. Add Flight\n");
        printf("2. Book Ticket\n");
        printf("3. Cancel Ticket\n");
        printf("4. Display Flight Details\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 1) {
```

```c
        if (choice == 1) {
            printf("Enter flight number: ");
            scanf("%s", flights[flight_count].flight_number);
            printf("Enter destination city: ");
            scanf(" %[^\n]", flights[flight_count].destination);
            printf("Enter number of available seats: ");
            scanf("%d", &flights[flight_count].available_seats);
            flight_count++;
        }
        else if (choice == 2) {
            char flight_number[10];
            printf("Enter flight number to book ticket: ");
            scanf("%s", flight_number);

            int found = 0;
            for (int i = 0; i < flight_count; i++) {
                if (strcmp(flights[i].flight_number, flight_number) == 0) {
                    if (flights[i].available_seats > 0) {
                        flights[i].available_seats--;
                        printf("Ticket booked successfully. Available seats: %d\n", flights[i].available_seats);
                    } else {
                        printf("No available seats on this flight.\n");
```

```c
        for (int i = 0; i < flight_count; i++) {
            if (strcmp(flights[i].flight_number, flight_number) == 0) {
                } else {
                    printf("No available seats on this flight.\n");
                }
                found = 1;
                break;
            }
        }
        if (!found) {
            printf("Flight not found.\n");
        }
    }
    else if (choice == 3) {
        char flight_number[10];
        printf("Enter flight number to cancel ticket: ");
        scanf("%s", flight_number);

        int found = 0;
        for (int i = 0; i < flight_count; i++) {
            if (strcmp(flights[i].flight_number, flight_number) == 0) {
                flights[i].available_seats++;
                printf("Ticket canceled successfully. Available seats: %d\n", flights[i].available_seats);
```

```c
                printf("Ticket canceled successfully. Available seats: %d\n", fli
                found = 1;
                break;
            }
        }
        if (!found) {
            printf("Flight not found.\n");
        }
    }
    else if (choice == 4) {
        char destination[50];
        printf("Enter destination city to display flight details: ");
        scanf(" %[^\n]", destination);

        int found = 0;
        for (int i = 0; i < flight_count; i++) {
            if (strcmp(flights[i].destination, destination) == 0) {
                printf("Flight Number: %s\n", flights[i].flight_number);
                printf("Destination: %s\n", flights[i].destination);
```

```
76              printf("Enter destination city to display flight details: ");
77              scanf(" %[^\n]", destination);
78
79              int found = 0;
80              for (int i = 0; i < flight_count; i++) {
81                  if (strcmp(flights[i].destination, destination) == 0) {
82                      printf("Flight Number: %s\n", flights[i].flight_number);
83                      printf("Destination: %s\n", flights[i].destination);
84                      printf("Available Seats: %d\n", flights[i].available_seats);
85                      found = 1;
86                  }
87              }
88              if (!found) {
89                  printf("No flights found to this destination.\n");
90              }
91          }
92      } while (choice != 5);
93
94      return 0;
95  }
96
```

```
PS D:\c progrms coding> gcc structass5.c
PS D:\c progrms coding> ./a

Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flight Details
5. Exit
Enter your choice: 1
Enter flight number: 123
Enter destination city: pattaya
Enter number of available seats: 2

Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flight Details
5. Exit
Enter your choice: 2
Enter flight number to book ticket: 123
Ticket booked successfully. Available seats: 1

Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flight Details
5. Exit
```

```
4. Display Flight Details
5. Exit
Enter your choice: 3
Enter flight number to cancel ticket: 123
Ticket canceled successfully. Available seats: 2

Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flight Details
5. Exit
Enter your choice: 2
Enter flight number to book ticket: 123
Ticket booked successfully. Available seats: 1

Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flight Details
5. Exit
Enter your choice: 4
Enter destination city to display flight details: pattaya
Flight Number: 123
Destination: pattaya
Available Seats: 1

Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
```

```
Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flight Details
5. Exit
Enter your choice: 4
Enter destination city to display flight details: pattaya
Flight Number: 123
Destination: pattaya
Available Seats: 1

Flight Reservation System Menu:
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flight Details
5. Exit
```