PROGRAM 5

Objective:- To create a simple C program that uses pointers to manage students records. The program demonstrate pointers operations, including pointer arithmetic, dereferencing.

Project description :->

The program allow the user to :

1. Enter student details (Name, Age, Grade)
2. Display student details.
3. Modify student details.

Code:

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

// Define a structure to store student details
typedef struct {
    char name[50];
    int age;
    char grade;
} Student;

// Function declarations
void enterDetails(Student *s);
void displayDetails(const Student *s);
void modifyDetails(Student *s);

int main() {
    Student student;  // Create a Student structure
    Student *ptr = &student;  // Pointer to the structure
    int choice;
```

```
    do {
        printf("\nMenu:\n");
        printf("1. Enter Student Details\n");
        printf("2. Display Student Details\n");
        printf("3. Modify Student Details\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                enterDetails(ptr);
                break;
            case 2:
                displayDetails(ptr);
                break;
            case 3:
                modifyDetails(ptr);
                break;
            case 4:
                printf("Exiting the program.\n");
                break;
            default:
                printf("Invalid choice. Try again.\n");
        }
    } while (choice != 4);

    return 0;
}
```

```
// Function to enter student details
void enterDetails(Student *s) {
    printf("Enter name: ");
    scanf(" %[^\n]s", s->name); // Read string with spaces
    printf("Enter age: ");
    scanf("%d", &s->age);
    printf("Enter grade: ");
    scanf(" %c", &s->grade); // Space before %c to consume newline
}

// Function to display student details
void displayDetails(const Student *s) {
    printf("\nStudent Details:\n");
    printf("Name: %s\n", s->name);
    printf("Age: %d\n", s->age);
    printf("Grade: %c\n", s->grade);
}

// Function to modify student details
void modifyDetails(Student *s) {
    int choice;
    printf("\nWhat do you want to modify?\n");
    printf("1. Name\n2. Age\n3. Grade\nEnter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            printf("Enter new name: ");
            scanf(" %[^\n]s", s->name);
```

```
            break;
        case 2:
            printf("Enter new age: ");
            scanf("%d", &s->age);
            break;
        case 3:
            printf("Enter new grade: ");
            scanf(" %c", &s->grade);
            break;
        default:
            printf("Invalid choice.\n");
    }
}
```

Outputs:-

Run 1: Entering and Displaying the student details

Input:

```
Menu:
1. Enter Student Details
2. Display Student Details
3. Modify Student Details
4. Exit
Enter your choice: 1
Enter name: Yash
Enter age: 19
Enter grade: A

Menu:
1. Enter Student Details
2. Display Student Details
3. Modify Student Details
4. Exit
Enter your choice: 2
```

Output:

```
Student Details:
Name: Yash
Age: 19
Grade: A
```

Run 2: Modifying Student Details Input: Output:

```
Menu:
1. Enter Student Details
2. Display Student Details
3. Modify Student Details
4. Exit
Enter your choice: 1
Enter name: Sangram
Enter age: 20
Enter grade: B

Menu:
1. Enter Student Details
2. Display Student Details
3. Modify Student Details
4. Exit
Enter your choice: 3

What do you want to modify?
1. Name
2. Age
3. Grade
Enter your choice: 3
Enter new grade: A

Menu:
1. Enter Student Details
2. Display Student Details
3. Modify Student Details
4. Exit
Enter your choice: 2
```

```
Menu:
1. Enter Student Details
2. Display Student Details
3. Modify Student Details
4. Exit
Enter your choice: 2

Student Details:
Name: Sangram
Age: 20
Grade: A
```

Run 3: Invalid Choice and Exciting

Input:

```
Menu:
1. Enter Student Details
2. Display Student Details
3. Modify Student Details
4. Exit
Enter your choice: 4
```

Output:

```
Exiting the program.
```

PROGRAM 6

Develop a C Program to Manage and Compare Employee Records in a Company Using Structures

Description: This project aims to design and implement a program that allows storing and displaying the details of multiple employees in a company using structures. Additionally, the program will include functionality to compare the records of two employees based on specified attributes (e.g., salary, age, or ID) and display the result of the comparison.

Code:

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

// Structure definition for Employee
struct Employee {
    int id;
    char name[50];
    int age;
    float salary;
};

// Function prototypes
void inputEmployeeDetails(struct Employee *emp);
void displayEmployeeDetails(const struct Employee *emp);
void compareEmployees(const struct Employee *emp1, const struct Employee *emp2);

int main() {
    int n;

    printf("Enter the number of employees: ");
    scanf("%d", &n);
```

```
if (n <= 0) {
    printf("Invalid number of employees. Exiting program.\n");
    return 1;
}

struct Employee employees[n];

// Input employee details
for (int i = 0; i < n; i++) {
    printf("\nEnter details for employee %d:\n", i + 1);
    inputEmployeeDetails(&employees[i]);
}

// Display employee details
printf("\nEmployee Details:\n");
for (int i = 0; i < n; i++) {
    printf("\nEmployee %d:\n", i + 1);
    displayEmployeeDetails(&employees[i]);
}

// Compare two employees
int emp1Index, emp2Index;
printf("\nEnter the indices of two employees to compare (1 to %d): ", n);
scanf("%d %d", &emp1Index, &emp2Index);

if (emp1Index < 1 || emp1Index > n || emp2Index < 1 || emp2Index > n) {
    printf("Invalid indices entered. Exiting program.\n");
    return 1;
}
```

```
    compareEmployees(&employees[emp1Index - 1], &employees[emp2Index - 1]);

    return 0;
}

void inputEmployeeDetails(struct Employee *emp) {
    printf("Enter ID: ");
    scanf("%d", &emp->id);

    printf("Enter Name: ");
    scanf("%s", emp->name);

    printf("Enter Age: ");
    scanf("%d", &emp->age);

    printf("Enter Salary: ");
    scanf("%f", &emp->salary);
}

void displayEmployeeDetails(const struct Employee *emp) {
    printf("ID: %d\n", emp->id);
    printf("Name: %s\n", emp->name);
    printf("Age: %d\n", emp->age);
    printf("Salary: %.2f\n", emp->salary);
}

void compareEmployees(const struct Employee *emp1, const struct Employee *emp2) {
    printf("\nComparing Employee %d and Employee %d:\n", emp1->id, emp2->id);
```

```
    if (emp1->salary > emp2->salary) {
        printf("Employee %d has a higher salary than Employee %d.\n", emp1->id, emp2->id);
    } else if (emp1->salary < emp2->salary) {
        printf("Employee %d has a lower salary than Employee %d.\n", emp1->id, emp2->id);
    } else {
        printf("Both employees have the same salary.\n");
    }

    if (emp1->age > emp2->age) {
        printf("Employee %d is older than Employee %d.\n", emp1->id, emp2->id);
    } else if (emp1->age < emp2->age) {
        printf("Employee %d is younger than Employee %d.\n", emp1->id, emp2->id);
    } else {
        printf("Both employees are of the same age.\n");
    }

    if (emp1->id > emp2->id) {
        printf("Employee %d has a higher ID than Employee %d.\n", emp1->id, emp2->id);
    } else if (emp1->id < emp2->id) {
        printf("Employee %d has a lower ID than Employee %d.\n", emp1->id, emp2->id);
    } else {
        printf("Both employees have the same ID.\n");
    }
}
```

Output:

User input-

```
Enter the number of employees: 2

Enter details for employee 1:
Enter ID: 201
Enter Name: Sangram
Enter Age: 32
Enter Salary: 65000

Enter details for employee 2:
Enter ID: 202
Enter Name: Pratap
Enter Age: 25
Enter Salary: 70000

Employee Details:

Employee 1:
ID: 201
Name: Sangram
Age: 32
Salary: 65000.00

Employee 2:
ID: 202
Name: Pratap
Age: 25
Salary: 70000.00

Enter the indices of two employees to compare (1 to 2): 1 2
```

Output-

```
Comparing Employee 201 and Employee 202:
Employee 201 has a lower salary than Employee 202.
Employee 201 is older than Employee 202.
Employee 201 has a lower ID than Employee 202.
```