C Test – Examination

1 What is Pointer ? Explain With Example

A pointer is a variable that stores the address of another variable.

Instead of holding a value directly, it holds the memory location of a variable.

Declration

```
   data_type *pointer_name;
```

* is used to declare a pointer.

data_type is the type of variable the pointer will point to

Initialization

```
int s = 10;

int *p = &s;
```

- &x gives the address of x, which is stored in pointer p

Example

```
#include <stdio.h>

int main() {

   int s =20 ;

   int *ptr;

   ptr = &s;  // Pointer initialized with address of x

     printf("Value of s = %d\n", s);

     printf("Address of s = %p\n", &s);

     printf("Value stored in ptr (address) = %p\n", ptr);

     printf("Value pointed by ptr = %d\n", *ptr);

   return 0;

}
```

2) Difference between C and C++

Point	C	C++
1.	Procedural programming language	Object-oriented + procedural language
2.	Does not support OOP	Supports object-oriented programming
3.	Limited code reuse	Code reuse via classes and inheritance
4.	No function overloading	Supports function overloading
5.	printf() and scanf()	cout and cin

3) Finding Duplicate Element Form array

```
#include <stdio.h>
int main() {
    int arr[100], n, i, j;

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

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

    printf("Duplicate elements are: ");
    for(i = 0; i < n; i++) {
        for(j = i + 1; j < n; j++) {
            if(arr[i] == arr[j]) {
                printf("%d ", arr[i]);
                break;
            }
        }
    }
    return 0;
}
```

4) Create an employee Structure

Eid

Ename

Esalary

Follwing opertion

1) Add employee form user input
2) Show all employee

All entry done file handling

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


struct employee {
    int eid;
    char ename[50];
    float esalary;
};
```

```
int main() {
    struct employee emp;
    FILE *fp;
    int choice, n, i;

    while (1) {
        printf("\nEmployee Management System\n");
        printf("1. Add Employees\n");
        printf("2. Show All Employees\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                fp = fopen("employees.txt", "a");  // append mode, text file
                if (fp == NULL) {
                    printf("Error opening file!\n");
                    return 1;
                }

                printf("How many employees do you want to add? ");
                scanf("%d", &n);
                for (i = 0; i < n; i++) {
                    printf("\nEnter details for employee %d\n", i + 1);
                    printf("Employee ID: ");
                    scanf("%d", &emp.eid);
                    printf("Employee Name: ");
                    scanf("%s", emp.ename);
                    printf("Employee Salary: ");
                    scanf("%f", &emp.esalary);

                    fprintf(fp, "%d %s %.2f\n", emp.eid, emp.ename, emp.esalary);
                }

                fclose(fp);
                printf("Employees added successfully!\n");
                break;

            case 2:
                fp = fopen("employees.txt", "r");
                if (fp == NULL) {
                    printf("No employee records found.\n");
                } else {
                    printf("\n=== Employee Records ===\n");
                    i = 0;
```

```
                    while (fscanf(fp, "%d %s %f", &emp.eid, emp.ename, &emp.esalary) == 3) {
                        printf("\nEmployee %d\n", ++i);
                        printf("ID     : %d\n", emp.eid);
                        printf("Name   : %s\n", emp.ename);
                        printf("Salary : %.2f\n", emp.esalary);
                    }
                    if (i == 0)
                        printf("No records to display.\n");

                    fclose(fp);
                }
                break;

            case 3:
                printf("Ending of the program\n");
                exit(0);
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
```

End of the exam