Class Lecture 10

❖ C program to implement the queue operations using an array.

Problem Solution

1. Use three functions for three operations insert, delete, and display.
2. Use the switch statement to access these functions.
3. Exit.

Program/Source Code

```
/*
 * C Program to Implement a Queue using an Array
 */
#include <stdio.h>

#define MAX 50

void insert();
void delete();
void display();
int queue_array[MAX];
int rear = - 1;
int front = - 1;
main()
{
    int choice;
    while (1)
    {
        printf("1.Insert element to queue \n");
        printf("2.Delete element from queue \n");
        printf("3.Display all elements of queue \n");
        printf("4.Quit \n");
        printf("Enter your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
            insert();
            break;
            case 2:
            delete();
            break;
            case 3:
```

```
            display();
            break;
            case 4:
            exit(1);
            default:
            printf("Wrong choice \n");
        } /* End of switch */
    } /* End of while */
} /* End of main() */

void insert()
{
    int add_item;
    if (rear == MAX - 1)
    printf("Queue Overflow \n");
    else
    {
        if (front == - 1)
        /*If queue is initially empty */
        front = 0;
        printf("Inset the element in queue : ");
        scanf("%d", &add_item);
        rear = rear + 1;
        queue_array[rear] = add_item;
    }
} /* End of insert() */

void delete()
{
    if (front == - 1 || front > rear)
    {
        printf("Queue Underflow \n");
        return ;
    }
    else
    {
        printf("Element deleted from queue is : %d\n", queue_array[front]);
        front = front + 1;
    }
} /* End of delete() */

void display()
{
    int i;
    if (front == - 1)
        printf("Queue is empty \n");
    else
    {
        printf("Queue is : \n");
        for (i = front; i <= rear; i++)
            printf("%d ", queue_array[i]);
        printf("\n");
    }
} /* End of display() */
```

Program Explanation

1. Ask the user for the operations like insert, delete, display, and exit.
2. According to the option entered, access its respective function using the switch statement. Use the variables front and rear to represent the first and last element of the queue.
3. In the function insert(), firstly check if the queue is full. If it is, then print the output as "Queue Overflow". Otherwise, take the number to be inserted as input and store it in the variable add_item. Copy the variable add_item to the array queue_array[] and increment the variable rear by 1.
4. In the function delete(), firstly check if the queue is empty. If it is, then print the output as "Queue Underflow". Otherwise print the first element of the array queue_array[] and decrement the variable front by 1.
5. In the function display(), using for loop print all the elements of the array starting from front to rear.
6. Exit.

Exercise

1. Implement a program to solve the Tower of Hanoi problem for n disks, and print the sequence of moves required to transfer all the disks from one peg to another while adhering to the rules.
2. Write down a program to Check Whether a Number is Palindrome or Not.