

Program 4

WAP to simulate the working of a circular queue of integers using an array. Provide the following operations: Insert, Delete & Display.

The program should print appropriate messages for queue empty and queue overflow conditions.

Observation :

WEEK-5
Ans WAP to simulate the working of a circular queue of integers using an array. Provide the following operations: Insert, Delete & Display. The program should print appropriate message for queue overflow conditions.

```
#include <stdio.h>
#include <stdlib.h>
#define N 5

int q[N];

void Enqueue(int item, int *rear, int *front, int q[]) {
    if (((*rear + 1) % N) == *front) {
        printf("Queue Overflow\n");
        return;
    }
    if (*front == -1) {
        *front = 0;
    }
    *rear = ((*rear + 1) % N);
    q[*rear] = item;
}

int Dequeue(int *front, int *rear, int q[]) {
    if (*front == -1 || *rear == -1) {
        return -1;
    }
    int temp = q[*front];
```

```
if (*front == *rear) {
    *rear = -1;
    *front = -1;
}
else {
    *front = ((*front + 1) % N);
}
return temp;
}

void displayQ(int front, int rear, int q[]) {
    if (front == -1 || rear == -1) {
        printf("Queue is empty\n");
        return;
    }
    printf("Contents of Queue:\n");
    for (int i = front; i != rear; i = (i + 1) % N) {
        printf("%d ", q[i]);
    }
    printf("%d", q[rear]);
}
```

Code:

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

#define N 5

int q[N];

void Enqueue(int item, int *rear, int *front, int q[]) {
    if ((*rear+1)%N == *front) {
        printf("Queue Overflow \n");
        return;
    }
    if (*front == -1)
    {
        *front = 0;
    }

    *rear = ((*rear + 1)%N);
    q[*rear] = item;
}

int Dequeue(int *front, int *rear, int q[]) {
    if (*front == -1 && *rear == -1) {
        return -1;
    }
    int temp = q[*front];
    if(*front == *rear){
        *rear = -1;
        *front = -1;
    }
    else{
        *front = ((*front+1)%N);
    }
    return temp;
}

void displayQ(int front, int rear, int q[]) {
    if (front == -1 && rear == -1) {
        printf("Queue is empty \n");
        return;
    }
    printf("Contents of Queue:\n");
    for (int i = front; i != rear; i=(i+1)%N) {
        printf("%d ", q[i]);
    }
    printf("%d", q[rear]);
}
```

```

}

int main() {
    int choice;
    int rear = -1;
    int front = -1;
    int item;
    for (;;) {
        printf("\nMenu:\n1: Enqueue\n2: Dequeue\n3: Display\n4: Exit\n");
        printf("Enter your choice:");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter the item to be inserted: ");
                scanf("%d", &item);
                Enqueue(item, &rear, &front , q);
                break;
            case 2:
                item = Dequeue(&front, &rear, q);
                if (item == -1)
                    printf("Queue is empty\n");
                else
                    printf("Item deleted = %d \n", item);
                break;
            case 3:
                displayQ(front, rear, q);
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice, please try again.\n");
        }
    }
    return 0;
}

```

Output:

```
PS C:\Users\satis> & 'c:\Users\satis\.vscode\extensions\w
tdin=Microsoft-MIEngine-In-jccpmqvw.xwf' '--stdout=Microso
d=Microsoft-MIEngine-Pid-wxivilue.dcv' '--dbgExe=C:\msys64

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 1

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 4

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 6

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 8

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:3
Contents of Queue:
1 4 6 8
```

```
Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:2
Item deleted = 1

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:2
Item deleted = 4

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:3
Contents of Queue:
6 8
Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 9

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 3
```

```
Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 2

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:1
Enter the item to be inserted: 5
Queue Overflow

Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:3
Contents of Queue:
6 8 9 3 2
Menu:
1: Enque
2: Deque
3: Display
4: Exit
Enter your choice:
```