

EXPERIMENT – 09

Aim : Program for Circular Queue implementation using dynamic memory allocation.

Procedure : A Circular Queue is an extended version of a normal queue where the last element of the queue is connected to the first element of the queue forming a circle.

Program in C :

```
#include<stdio.h>

#include <stdlib.h>

typedef struct list
{
    int data;
    struct list *link;
} NODE;

typedef struct Queue
{
    struct list *front, *rear;
} QUEUE;

void Create_Queue(QUEUE *q)
{
    q->front = NULL;
    q->rear = NULL;
}

void enQueue(QUEUE *q, int value)
```

```

{
    NODE *temp = (NODE *)malloc(sizeof(NODE));
    temp->data = value;
    if (q->front == NULL)
        q->front = temp;
    else
        q->rear->link = temp;
    q->rear = temp;
    q->rear->link = q->front;
}

int deQueue(QUEUE *q)
{
    NODE *temp;
    int value;
    if (q->front == NULL)
    {
        printf("Queue is empty");
        return -1;
    }
    if (q->front == q->rear)
    {
        value = q->front->data;
        free(q->front);
        q->front = NULL;
    }
}

```

```

        q->rear = NULL;
    }
    else
    {
        temp = q->front;
        value = temp->data;
        q->front = q->front->link;
        q->rear->link = q->front;
        free(temp);
    }
    return value;
}

void display(QUEUE *q)
{
    NODE *temp = q->front;
    if (temp == NULL)
    {
        printf("\nQUEUE is EMPTY.");
        return;
    }
    printf("\nElements in Circular Queue are: ");
    while (temp->link != q->front)
    {
        printf("%d ", temp->data);

```

```

        temp = temp->link;
    }
    printf("%d", temp->data);
}
void main()
{
    printf("\nName - Aditya Raj");
    QUEUE q;
    int value, y, ch;
    Create_Queue(&q);
    while (1)
    {
        printf("\n1 : ADD");
        printf("\n2 : DELETE");
        printf("\n3 : Display");
        printf("\n4 : Exit");
        printf("\n\nEnter your choice : ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:
                printf("\nEnter value : ");
                scanf("%d", &value);
                enqueue(&q, value);

```

```
        break;
    case 2:
        y = deQueue(&q);
        if (y != -1)
            printf("Deleted item is %d", y);
        break;
    case 3:
        display(&q);
        break;
    case 4:
        exit(0);
    }
}
}
```

Ouput :

Name - Aditya Raj

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 1

Enter value : 11

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 1

Enter value : 22

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 3

Elements in Circular Queue are: 11 22

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 2

Deleted item is 11

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 1

Enter value : 11

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 3

Elements in Circular Queue are: 22 11

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 2

Deleted item is 22

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 2

Deleted item is 11

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 3

QUEUE is EMPTY.

1 : ADD
2 : DELETE
3 : Display
4 : Exit

Enter your choice : 4

PS D:\C Program\DSA> █