DSA LAB ASSIGNMENT

(SIMPLE QUEUE&CIRCULAR QUEUE)

1. Simple Queue:

#include <stdio.h>

#include <stdlib.h>

struct Queue

{

    int size;

    int front;

    int rear;

    int \*Q;

};

void create(struct Queue \*q, int size)

{

    q->size = size;

    q->front = q->rear = -1;

    q->Q = (int \*)malloc(q->size \* sizeof(int));

}

void enqueue(struct Queue \*q, int x)

{

    if (q->rear == q->size - 1)

        printf("Queue is Full");

    else

    {

        q->rear++;

        q->Q[q->rear] = x;

    }

}

int dequeue(struct Queue \*q)

{

    int x = -1;

    if (q->front == q->rear)

        printf("Queue is Empty\n");

    else

    {

        q->front++;

        x = q->Q[q->front];

    }

    return x;

}

void Display(struct Queue q)

{

    int i;

    for (i = q.front + 1; i <= q.rear; i++)

        printf("%d ", q.Q[i]);

    printf("\n");

}

int main()

{

    struct Queue q;

    create(&q, 5);

    enqueue(&q, 10);

    enqueue(&q, 20);

    enqueue(&q, 30);

    enqueue(&q,40);

    enqueue(&q,50);

    printf("the Queue is : ");

    Display(q);

    dequeue(&q);

    dequeue(&q);

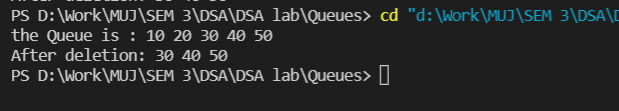
    printf("After deletion: ");

    Display(q);

    return 0;

}

Output-



1. Circular Queue :

#include <stdio.h>

#include <stdlib.h>

struct Queue

{

    int size;

    int front;

    int rear;

    int \*Q;

};

void create(struct Queue \*q, int size)

{

    q->size = size;

    q->front = q->rear = 0;

    q->Q = (int \*)malloc(q->size \* sizeof(int));

}

void enqueue(struct Queue \*q, int x)

{

    if ((q->rear + 1) % q->size == q->front)

        printf("Queue is Full");

    else

    {

        q->rear = (q->rear + 1) % q->size;

        q->Q[q->rear] = x;

    }

}

int dequeue(struct Queue \*q)

{

    int x = -1;

    if (q->front == q->rear)

        printf("Queue is Empty\n");

    else

    {

        q->front = (q->front + 1) % q->size;

        x = q->Q[q->front];

    }

    return x;

}

void Display(struct Queue q)

{

    int i = q.front + 1;

    do

    {

        printf("%d ", q.Q[i]);

        i = (i + 1) % q.size;

    } while (i != (q.rear + 1) % q.size);

    printf("\n");

}

int main()

{

    struct Queue q;

    create(&q, 7);

    enqueue(&q, 10);

    enqueue(&q, 20);

    enqueue(&q, 30);

    enqueue(&q, 40);

    enqueue(&q, 50);

    enqueue(&q, 60);

    printf("The Circular queue made is : ");

    Display(q);

    dequeue(&q);

    dequeue(&q);

    dequeue(&q);

    printf("After deletion the given circular Queue is : ");

    Display(q);

    return 0;

}

Output :

