Simple Queue

#include <stdio.h>

#include <stdlib.h>

struct Node

{

    int data;

    struct Node \*next;

} \*front = NULL, \*rear = NULL;

void enqueue(int x)

{

    struct Node \*t;

    t = (struct Node \*)malloc(sizeof(struct Node));

    if (t == NULL)

        printf("Queue is FUll\n");

    else

    {

        t->data = x;

        t->next = NULL;

        if (front == NULL)

            front = rear = t;

        else

        {

            rear->next = t;

            rear = t;

        }

    }

}

int dequeue()

{

    int x = -1;

    struct Node \*t;

    if (front == NULL)

        printf("Queue is Empty\n");

    else

    {

        x = front->data;

        t = front;

        front = front->next;

        free(t);

    }

    return x;

}

void Display()

{

    struct Node \*p = front;

    while (p)

    {

        printf("%d ", p->data);

        p = p->next;

    }

    printf("\n");

}

int main()

{

    enqueue(10);

    enqueue(20);

    enqueue(30);

    enqueue(40);

    enqueue(50);

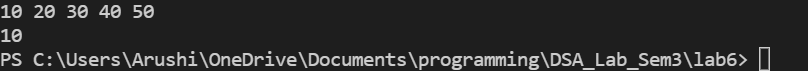
    Display();

    printf("%d ", dequeue());

    return 0;

}

Output:



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\n");

    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, 5);

    enqueue(&q, 10);

    enqueue(&q, 20);

    enqueue(&q, 30);

    enqueue(&q, 40);

    enqueue(&q, 50);

    enqueue(&q, 60);

    Display(q);

    printf("%d ", dequeue(&q));

    return 0;

}

Output:

