**Edwin Ario Abdiwijaya**

[**edwin.abdiwijaya@binus.ac.id**](mailto:edwin.abdiwijaya@binus.ac.id)

**Answer Sheet for Assignment 7**

**Answer for question 1 :**

// Edwin Ario Abdiwijaya

// edwin.abdiwijaya@binus.ac.id

// Answer for question number 1

#include <stdio.h>

#include <stdlib.h>

struct Node

{

    int data;

    struct Node \*link;

};

struct Queue

{

    struct Node \*front, \*rear;

};

void insertQueue(Queue \*que, int value)

{

    struct Node \*temp = new Node;

    temp->data = value;

    if (que->front == NULL)

        que->front = temp;

    else

        que->rear->link = temp;

    que->rear = temp;

    que->rear->link = que->front;

}

int deleteQueue(Queue \*que)

{

    if (que->front == NULL)

    {

        printf("Queue is empty");

        return INT\_MIN;

    }

    int value;

    if (que->front == que->rear)

    {

        value = que->front->data;

        free(que->front);

        que->front = NULL;

        que->rear = NULL;

    }

    else

    {

        struct Node \*temp = que->front;

        value = temp->data;

        que->front = que->front->link;

        que->rear->link = que->front;

        free(temp);

    }

    return value;

}

void displayQueue(struct Queue \*que)

{

    struct Node \*temp = que->front;

    printf("\n");

    printf("Elements in Circular Queue are: ");

    while (temp->link != que->front)

    {

        printf("%d ", temp->data);

        temp = temp->link;

    }

    printf("%d", temp->data);

}

int main()

{

    Queue \*que = new Queue;

    que->front = que->rear = NULL;

    insertQueue(que, 1);

    insertQueue(que, 3);

    insertQueue(que, 5);

    displayQueue(que);

    printf("Deleted value = %d\n", deleteQueue(que));

    printf("Deleted value = %d\n", deleteQueue(que));

    displayQueue(que);

    insertQueue(que, 7);

    insertQueue(que, 9);

    displayQueue(que);

    return 0;

}