TITLE 47

Write a C program to implement the queue using a linked list

OBJECTIVE:

To implement the queue using a linked list

PROBLEM STATEMENT:

In this problem using linked list we implement the QUEUE.

ALGORITHM:

START

INPUT: input from the users

COMPUTATION: Computing Queue front and Queue rear using a linked list

DISPLAY: Displaying queue front and queue rear with the help of linked list

STOP

PROGRAM:

#include <stdio.h>

#include <stdlib.h>

struct QNode {

    int key;

    struct QNode\* next;

};

struct Queue {

    struct QNode \*front, \*rear;

};

struct QNode\* newNode(int k)

{

    struct QNode\* temp = (struct QNode\*)malloc(sizeof(struct QNode));

    temp->key = k;

    temp->next = NULL;

    return temp;

}

struct Queue\* createQueue()

{

    struct Queue\* q = (struct Queue\*)malloc(sizeof(struct Queue));

    q->front = q->rear = NULL;

    return q;

}

void enQueue(struct Queue\* q, int k)

{

    struct QNode\* temp = newNode(k);

    if (q->rear == NULL) {

        q->front = q->rear = temp;

        return;

    }

    q->rear->next = temp;

    q->rear = temp;

}

void deQueue(struct Queue\* q)

{

    if (q->front == NULL)

        return;

    struct QNode\* temp = q->front;

    q->front = q->front->next;

    if (q->front == NULL)

        q->rear = NULL;

    free(temp);

}

int main()

{

    struct Queue\* q = createQueue();

    enQueue(q, 10);

    enQueue(q, 20);

    deQueue(q);

    deQueue(q);

    enQueue(q, 30);

    enQueue(q, 40);

    enQueue(q, 50);

    deQueue(q);

    printf("Queue Front : %d \n", q->front->key);

    printf("Queue Rear : %d", q->rear->key);

    return 0;

}

CONCLUSION:

This program helps us to understand how linked list are used for implementing the queue.

OUTPUT:

Queue Front : 40

Queue Rear : 50