Experiment-14 : QUEUE Operations

14.Write a C Program To Implement QUEUE Operations Such As ENQUEUE , DEQUEUE And Display.

Code :

#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 100

typedef struct {

int data[MAX\_SIZE];

int front;

int rear;

} Queue;

void initializeQueue(Queue \*queue) {

queue->front = -1;

queue->rear = -1;

}

int isEmpty(Queue \*queue) {

return (queue->front == -1 && queue->rear == -1);

}

int isFull(Queue \*queue) {

return (queue->rear == MAX\_SIZE - 1);

}

void enqueue(Queue \*queue, int element) {

if (isFull(queue)) {

printf("Queue is full. Cannot enqueue element.\n");

return;

}

if (isEmpty(queue)) {

queue->front = 0;

queue->rear = 0;

} else {

queue->rear++;

}

queue->data[queue->rear] = element;

}

int dequeue(Queue \*queue) {

if (isEmpty(queue)) {

printf("Queue is empty. Cannot dequeue element.\n");

return -1;

}

int element = queue->data[queue->front];

if (queue->front == queue->rear) {

queue->front = -1;

queue->rear = -1;

} else {

queue->front++;

}

return element;

}

void displayQueue(Queue \*queue) {

if (isEmpty(queue)) {

printf("Queue is empty.\n");

return;

}

printf("Queue elements: ");

for (int i = queue->front; i <= queue->rear; i++) {

printf("%d ", queue->data[i]);

}

printf("\n");

}

int main() {

Queue queue;

initializeQueue(&queue);

enqueue(&queue, 10);

enqueue(&queue, 20);

enqueue(&queue, 30);

displayQueue(&queue);

int dequeuedElement = dequeue(&queue);

printf("Dequeued element: %d\n", dequeuedElement);

displayQueue(&queue);

return 0;

}

Output :

