**IMPLEMENTATION OF QUEUE USING ARRAY AND LINKED LIST.**

**Aim:**

The aim of the program is to execute

queue using array and linked list.

**Algorithm:**

1. Start
2. To enqueuer an element read the value
3. If rear is equal to MAX\_SIZE-1, print Queue is full
4. Otherwise if front is -1, set front to 0 , increment rear by 1 and assign the value to queue[rear].
5. To dequeuer an element if front is -1 print Queue is emprty and return 1
6. Otherwise assign element as queue[front], increment front by 1
7. End

**Program using array:**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 100

int queue[MAX\_SIZE];

int front = -1, rear = -1;

void enqueue(int value);

int dequeue();

void display();

int main() {

enqueue(10);

enqueue(20);

enqueue(30);

display();

dequeue();

display();

return 0;

}

void enqueue(int value) {

if (rear == MAX\_SIZE - 1) {

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

} else {

if (front == -1) {

front = 0;

}

rear++;

queue[rear] = value;

}

}

int dequeue() {

int element;

if (front == -1) {

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

return -1;

} else {

element = queue[front];

front++;

if (front > rear) {

front = rear = -1;

}

return element;

}

}

void display() {

if (front == -1) {

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

} else {

printf("Queue elements: ");

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

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

}

printf("\n");

}

}

**Output:**

Queue elements: 10 20 30

Queue elements: 20 30

**Program using linked list:**

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Node\* front = NULL;

struct Node\* rear = NULL;

void enqueue(int value);

int dequeue();

void display();

int main() {

enqueue(10);

enqueue(20);

enqueue(30);

display();

dequeue();

display();

return 0;

}

void enqueue(int value) {

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

newNode->data = value;

newNode->next = NULL;

if (rear == NULL) {

front = rear = newNode;

} else {

rear->next = newNode;

rear = newNode;

}

}

int dequeue() {

if (front == NULL) {

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

return -1;

} else {

struct Node\* temp = front;

int element = temp->data;

front = front->next;

free(temp);

if (front == NULL) {

rear = NULL;

}

return element;

}

}

void display() {

if (front == NULL) {

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

} else {

struct Node\* temp = front;

printf("Queue elements: ");

while (temp != NULL) {

printf("%d ", temp->data);

temp = temp->next;

}

printf("\n");

}

}

**Output:**

Queue elements: 10 20 30

Queue elements: 20 30

**Result:**

The output is verified successfully for the above program.