231501058 CS23231 – D a t a S t r u c t u r e s

**Ex. No.: 7**

**Implementation of Queue using Array and**

**Linked List Implementation**

**Date: 19/ 4/24**

**Write a C prog ram to imple me nt a Queue using Array an d lin ked L ist impleme ntation and e xe cute the following operation on stack .**

1. **Enqueue**
2. **Dequeue**
3. **Display the elements in a Queue**

**Algorithm:**

#include <stdio.h>

#include <stdlib.h>

struct node {

int data;

struct node\* next;

};

struct node\* head = NULL;

struct node\* tail = NULL;

void enqueue(int ele) {

struct node\* newnode = (struct node\*)malloc(sizeof(struct node)); if (newnode != NULL) {

newnode->data = ele;

newnode->next = NULL;

if (tail == NULL) {

head = tail = newnode;

} else {

tail->next = newnode;

tail = newnode;

}

}

}

int dequeue() {

if (head == NULL) {

printf("\nQueue Underflow\n");

return -1;

} else {

int dequeued = head->data;

struct node\* temp = head;

head = head->next;

if (head == NULL) {

tail = NULL;

}

free(temp);



**Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College**

. 29

231501058 CS23231 – D a t a S t r u c t u r e s

return dequeued;

}

}

int isEmpty() {

return head == NULL;

}

void display() {

struct node\* t = head;

while (t != NULL) {

printf("%d\t", t->data);

1. = t->next;

}

printf("\n");

}

int main() {

enqueue(10);

enqueue(20);

enqueue(30);

display();

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

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

printf("Is queue empty? %s\n", isEmpty() ? "Yes" : "No");

enqueue(40);

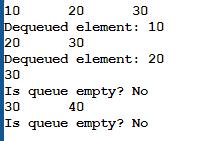
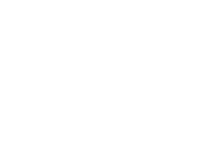
display();

printf("Is queue empty? %s\n", isEmpty() ? "Yes" : "No");

return 0;

}

**OUTPUT**



**Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College**

. 30