DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment – 3.1

Student Name: Anushikha das **UID: 21BCS7399**
Branch: CSE **Section/Group: 717-A**
Semester: 3rd **Date of Performance: 27/10/2022**
Subject Name: Data structure **SubjectCode:21CSH-211**

1. **Aim of the practical:** write a program to demonstrate the implementation of various operations on a queue represented using a linear linked list (linked queue).

2. **ALGORITHM:**

For insertion

1. Allocate space for new node ptr
2. Set ptr -> data = value
3. If front == NULL
4. Set front = rear= ptr
5. Set front->next = prt->next = NULL
6. Else, set rear ->next =ptr
7. Rear =ptr
8. Rear->next=NULL
9. Exit

For deletion

1. If front = NULL then underflow and exit
2. Else set ptr = front
3. front =front->next
4. Free(ptr)
5. exit

3. **Program Code:**

```
#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) {
    struct node * ptr;
    ptr = (struct node * ) malloc(sizeof(struct node));
    ptr - > data = value;
    ptr - > next = NULL;
    if ((front == NULL) && (rear == NULL)) {
        front = rear = ptr;
    } else {
        rear - > next = ptr;
        rear = ptr;
    }
    printf("Node is Inserted\n\n");
}
int dequeue() {
    if (front == NULL) {
        printf("\nUnderflow\n");
        return -1;
    } else {
        struct node * temp = front;
        int temp_data = front - > data;
        front = front - > next;
        free(temp);
        return temp_data;
```

```
    }
}

void display() {
    struct node * temp;
    if ((front == NULL) && (rear == NULL)) {
        printf("\nQueue is Empty\n");
    } else {
        printf("The queue is \n");
        temp = front;
        while (temp) {
            printf("%d--->", temp - > data);
            temp = temp - > next;
        }
        printf("NULL\n\n");
    }
}

int main() {
    int choice, value;
    printf("\nImplementation of Queue using Linked List\n");
    while (choice != 4) {
        printf("1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n");
        printf("\nEnter your choice : ");
        scanf("%d", & choice);
        switch (choice) {
            case 1:
                printf("\nEnter the value to insert: ");
                scanf("%d", & value);
                enqueue(value);
                break;
            case 2:
                printf("Popped element is :%d\n", dequeue());
                break;
            case 3:
                display();
                break;
            case 4:
```

```
                exit(0);
                break;
            default:
                printf("\nWrong Choice\n");
        }
    }
    return 0;
}
```

4. OUTPUT

```
Implementation of Queue using Linked List
1. Enqueue
2. Dequeue
3. Display
4. Exit

Enter your choice: 1

Enter the value to insert: 12
Node is Inserted

1.Enqueue
2.Dequeue
3.Display
4.Exit

Enter your choice: 1
```

```
Enter the value to insert: 45
Node is Inserted

1.Enqueue
2.Dequeue
3.Display
4.Exit

Enter your choice: 1

Enter the value to insert: 56
Node is Inserted

1.Enqueue
2.Dequeue
3.Display
4.Exit

Enter your choice : 3
The queue is
12--->45--->56--->NULL
```

Learning Outcomes (What I have learnt)

1. Learnt about queue and its types
2. Also learnt the various operations on queue