



Experiment: 3.1

Student Name: Rohit Panghal

Branch: CSE

Semester: 3rd

**Subject Name: Data Structures and
Algorithms**

UID: 21BCS9294

Section/Group: 902-A

Date: 01-11-2022

Subject code: 21CSH-211

AIM:

Write a program to demonstrate the implementation of various operations on a queue represented using a linear linked list (linked queue).

ALGORITHM:

Insert operation:

1. Allocate the space for the new node PTR.

2. SET PTR -> DATA = VAL

3. IF FRONT = NULL

 SET FRONT = REAR = PTR

 SET FRONT -> NEXT = REAR -> NEXT = NULL

ELSE

 SET REAR -> NEXT = PTR

 SET REAR = PTR

 SET REAR -> NEXT = NULL

[END OF IF]



Deletion:

1. IF FRONT = NULL

Write " Underflow " Go to Step 5

[END OF IF]

2. SET PTR = FRONT

3. SET FRONT = FRONT -> NEXT

4. FREE PTR

5.END

PROGRAM CODE:

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
struct node{
```

```
int data;
```

```
struct node *next;
```

```
};
```

```
struct node *front;
```

```
struct node *rear;
```

```
void insert();
```

```
void delete();
```

```
void display();
```

```
void main(){
```

```
int choice;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
while(choice != 4){  
    printf("\n1.insert an element\n2.Delete an element\n3.Display the  
    queue\n4.Exit\n");  
    printf("Enter your choice :");  
    scanf("%d",& choice);  
    switch(choice){  
    case 1:  
        insert();  
        break;  
    case 2:  
        delete();  
        break;  
    case 3:  
        display();  
        break;  
    case 4:  
        exit(0);  
        break;  
    default:  
        printf("\nEnter valid choice\n");  
    } }  
void insert(){  
    struct node *ptr;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
int item;

ptr = (struct node *) malloc (sizeof(struct node));

if(ptr == NULL){
printf("OVERFLOW\n");
return;
}

else{
printf("\nEnter value: ");
scanf("%d",&item);
ptr -> data = item;
if(front == NULL){
front = ptr;
rear = ptr;
front -> next = NULL;
rear -> next = NULL;
}
else{
rear -> next = ptr;
rear = ptr;
rear->next = NULL;
}}

void delete(){
struct node *ptr;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
if(front == NULL){  
    printf("UNDERFLOW\n");  
    return;  
}  
else{  
    ptr = front;  
    front = front -> next;  
    free(ptr);  
} }  
  
void display(){  
    struct node *ptr;  
    ptr = front;  
    if(front == NULL){  
        printf("Empty queue\n");  
    }  
    else{  
        printf("printing values: ");  
        while(ptr != NULL){  
            printf("%d ", ptr -> data);  
            ptr = ptr -> next;  
        }  
    }
```

OUTPUT:

```
1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice :1

Enter value: 2

1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice :1

Enter value: 4

1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice :3
printing values: 2 4
1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice :|
```

LEARNING OUTCOME:

1. Learned about queue Representation.
2. About linked list.
3. About various operations in queue.