

Mechi Multiple Campus

(Tribhuvan University)

Bhadrapur, Jhapa



Lab Report of

Data Structures and Algorithm (CACS-201)

Implementation of Linear Queue

Faculty of Humanities & Social Sciences

Tribhuvan University

Kritipur, Nepal

Submitted By

Name: Santosh Bhandari

Roll No: 58

Submitted To

Mechi Multiple Campus

Department of Bachelor in Computer Application

Bhadrapur, Jhapa, Nepal

Introduction to Queue

Definition

Queue is a linear collection of data in which data are inserted from one end called as 'rear' and deleted from another end called as 'front.' It follows FIFO (First In First Out) order i.e. ~~that~~ the data inserted as the first will be the first one to be removed.

Applications

- i) Print Queue
- ii) Playlist in Media player
- iii) Reservation or Booking System
- iv) CPU Scheduling
- v) Traversal in Graph

Terminologies

i) Enqueue

The process of inserting data in the queue.

ii) Dequeue

The process of deleting data from the queue.

iii) Peek

The process of identifying the data to be deleted next.

iv) Full Queue

The condition in which all the location of the queue contains data.

v) Empty Queue

The condition in which Queue doesn't contain any data.

vi) Overflow

The result of inserting data in a full queue.

vii) Underflow

The result of deleting data from an empty queue.

Operations

i) Enqueue Operation

Adding the data into the Queue is referred to as Enqueue operation.

ii) Dequeue Operation

Deleting the data from the Queue is called dequeue operation.

iii) Peek or Display Operation

Display operation involves returning the element which is present in the Queue without deleting it.

Algorithm to insert and delete data from the linear Queue

Enqueue Operation Algorithm

- ① [Check if the Queue is full or not]
if $REAR = MAX - 1$
 Print OVERFLOW and EXIT
- ② [Update the Value of FRONT and REAR]
if $REAR = -1$
 then Set $REAR = 0$ and $FRONT = 0$
else
 $REAR = REAR + 1$
- ③ [Insert data at the REAR of Queue]
 $QUEUE[REAR] = \text{newdata}$
- ④ Exit

Dequeue Operation Algorithm

- ① [Check if the queue is empty or not]
if $REAR = -1$
 then print UNDERFLOW and Exit
- ② [Delete the Data]
 Delete $QUEUE[FRONT]$
- ③ [Update the value of FRONT and REAR]
if $REAR = FRONT$
 then Set $REAR = -1$
 $FRONT = -1$
else
 $FRONT = FRONT + 1$
- ④ Exit

Program Code

```
#include<stdio.h>
void enqueue();
void dequeue();
void display();
int queue[3],front=-1,rear=-1,max=3;
void main(){
    top:
    printf("\n***Option***\n1.Insert Data in Queue\n2.Remove Data From
    Queue\n3.Display Data of Queue\n4.Exit\n\nSelect Your Option(1,2,3): ");
    int n;
    scanf("%d",&n);
    switch(n){
        case 1:
            enqueue();
            goto top;
        case 2:
            dequeue();
            goto top;
        case 3:
            display();
            goto top;
        case 4:
            exit(0);
        default:
            printf("Wrong Entry.");
            goto top;
    }
}
void enqueue(){
    if(rear==(max-1))
        printf("OVERFLOW");
    else {
        if(rear== -1)
            rear=front=0;
        else
            rear++;
        printf("Enter a Data: ");
        scanf("%d",&queue[rear]);
        printf("%d is Inserted in Queue",queue[rear]);
    }
}
void dequeue(){
    if(rear== -1)
        printf("UNDERFLOW");
    else{
        printf("%d Deleted from Queue.",queue[front]);
        if(front==rear)
            front=rear=-1;
    }
}
```

```

        else
            front++;
    }
}
void display(){
    if(rear==-1)
        printf("Queue is Empty.");
    else{
        int i;
        printf("Data on Queue: ");
        for(i=front;i<=rear;i++)
            printf("%d\t",queue[i]);
    }
}
}

```

Output of the Program

```

***Option***
1.Insert Data in Queue
2.Remove Data From Queue
3.Display Data of Queue
4.Exit

Select Your Option(1,2,3): 1
Enter a Data: 10
10 is Inserted in Queue
***Option***
1.Insert Data in Queue
2.Remove Data From Queue
3.Display Data of Queue
4.Exit

Select Your Option(1,2,3): 1
Enter a Data: 20
20 is Inserted in Queue
***Option***
1.Insert Data in Queue
2.Remove Data From Queue
3.Display Data of Queue
4.Exit

Select Your Option(1,2,3): 1
Enter a Data: 30
30 is Inserted in Queue

```


Option

- 1.Insert Data in Queue
- 2.Remove Data From Queue
- 3.Display Data of Queue
- 4.Exit

Select Your Option(1,2,3): 3

Data on Queue: 10 20 30

Option

- 1.Insert Data in Queue
- 2.Remove Data From Queue
- 3.Display Data of Queue
- 4.Exit

Select Your Option(1,2,3): 1

OVERFLOW

Option

- 1.Insert Data in Queue
- 2.Remove Data From Queue
- 3.Display Data of Queue
- 4.Exit

Select Your Option(1,2,3): 2

10 Deleted from Queue.

Option

- 1.Insert Data in Queue
- 2.Remove Data From Queue
- 3.Display Data of Queue
- 4.Exit

Select Your Option(1,2,3): 2

20 Deleted from Queue.

Option

- 1.Insert Data in Queue
- 2.Remove Data From Queue
- 3.Display Data of Queue
- 4.Exit

Select Your Option(1,2,3): 2

30 Deleted from Queue.

Option

1. Insert Data in Queue
2. Remove Data From Queue
3. Display Data of Queue
4. Exit

Select Your Option(1,2,3): 3
Queue is Empty.

Option

1. Insert Data in Queue
2. Remove Data From Queue
3. Display Data of Queue
4. Exit

Select Your Option(1,2,3): 2
UNDERFLOW

Conclusion

Hence, Queue is a linear collection of data in which data are inserted from one end called 'rear' and deleted from another end called front. In Queue the first data inserted will be the first one to be deleted.

Linear Queue is a type of linear data structure that contains the element in a sequential manner. It doesn't allow to insert the new data if the last location of queue has data even if other location is empty.
