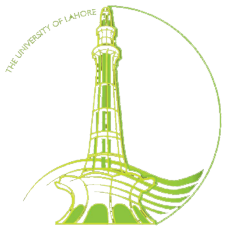

THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

# Data Structure And Algorithum

## Lab Report


| | |
|---|---|
| Name: | Ch Hasnain Zafar |
| Registration #: | SEU-F16-104 |
| Lab Report #: | 02 |
| Dated: | 12-03-2018 |
| Submitted To: | Mr. Usman Ahmed |

The University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

# Experiment # 1
# Queue Implementation Using Array

**Objective**
To understand and implement the Queue Problem.

**Software Tool**
1. DEV C++

## 1 Theory

structure can be implemented using one dimensional array. But, queue implemented using array can store only fixed number of data values. The implementation of queue data structure using array is very simple, just define a one dimensional array of specific size and insert or delete the values into that array by using FIFO (First In First Out) principle with the help of variables 'front' and 'rear'. Initially both 'front' and 'rear' are set to -1. Whenever, we want to insert a new value into the queue, increment 'rear' value by one and then insert at that position. Whenever we want to delete a value from the queue, then increment 'front' value by one and then display the value at 'front' position as deleted element.
Step 1: Include all the header files which are used in the program and define a constant 'SIZE' with specific value.
Step 2: Declare all the user defined functions which are used in queue implementation.
Step 3: Create a one dimensional array with above defined SIZE (int queue[SIZE])
Step 4: Define two integer variables 'front' and 'rear' and initialize both with '-1'. (int front = -1, rear = -1)
Step 5: Then implement main method by displaying menu of operations list and make suitable function calls to perform operation selected by the user on queue.

## 2 Task

### 2.1 Procedure: Task 1

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
#define SIZE 5
using namespace std;
int q[SIZE],front=0,rear=0;
void enqueue()
{
int no;
if (rear==SIZE && front==0)
cout<<"queue is full";
else
{
cout<<"enter the num:";
cin>>no;
q[rear]=no;
}
rear++;
}
void dequeue()
{
int no,i;
if (front==rear)
cout<<"queue is empty";
else
{
no=q[front];
front++;
cout<<"\n"<<no<<" -removed from the queue\n";
}
}
void display()
{
int i,temp=front;
```

```
if (front==rear)
cout<<"the queue is empty";
else
{
cout<<"\n element in the queue:";
for(i=temp;i<rear;i++)
{
cout<<q[i]<<" ";
}
}
}
int main()
{
int ch;

while(1)
{
cout<<"\n 1. add element";
cout<<"\n 2. remove element";
cout<<"\n 3.display";
cout<<"\n 4.exit";
cout<<"\n enter your choice:";
cin>>ch;
//clrscr();
switch(ch)
{
case 1:
enqueue();
break;
case 2:
dequeue();
break;
case 3:
display();
break;
case 4:
exit(0);
default:
cout<<"\n invalid choice";
}
```

```
}
}
```