



Data Structures and algorithms (CS09203)

Lab Report

Name: Irtaza Kashir Raja
Registration #: SEU-F16-127
Lab Report #: 01
Dated: 16-04-2018
Submitted To: Mr. Usman Ahmed

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

Experiment # 2

Queue with Array implementation

Objective

The objective of this session is to understand the various operations on queues using array structure in C++. **Software Tool**

1.To achieve the goals of objectives, I use Code Blocks with GCC compiler

1 Theory

Queue using Array: -

This manual discusses an important data structure, called a queue. The idea of a queue in computer science is the same as the idea of the queues to which you are accustomed in everyday life. There are queues of customers in a bank or in a grocery store and queues of cars waiting to pass through a tollbooth. Similarly, because a computer can send a print request faster than a printer can print, a queue of documents is often waiting to be printed at a printer. The general rule to process elements in a queue is that the customer at the front of the queue is served next and that when a new customer arrives, he or she stands at the end of the queue. That is, a queue is a First In First Out data structure.

A queue is a set of elements of the same type in which the elements are added at one end, called the back or rear, and deleted from the other end, called the front. For example, consider a line of customers in a bank, wherein the customers are waiting to withdraw/deposit money or to conduct some other business. Each new customer gets in the line at the rear. Whenever a teller is ready for a new customer, the customer at the front of the line is served.

The rear of the queue is accessed whenever a new element is added to the queue, and the front of the queue is accessed whenever an element is deleted from the queue. As in a stack, the middle elements of the queue are inaccessible, even if the queue elements are stored in an array.

Queue: A data structure in which the elements are added at one end, called the rear, and deleted from the other end, called the front; a First-In-First-

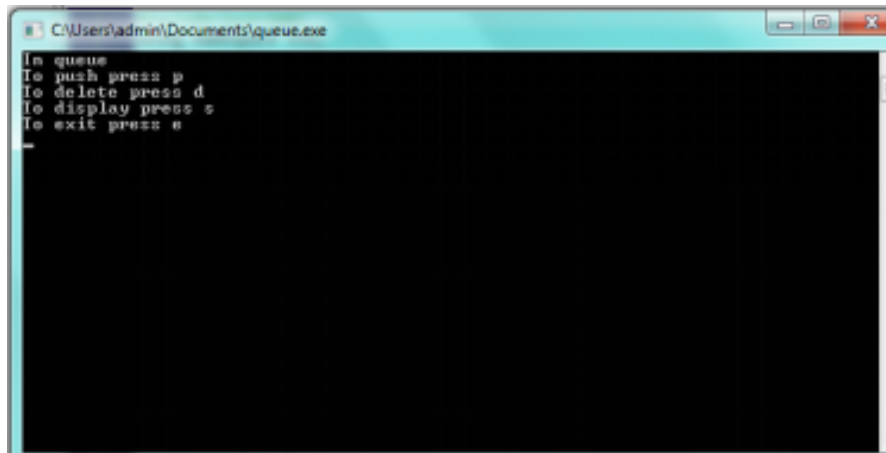


Figure 1: queue

Out (FIFO) data structure.

2 Task

2.1 Task 1

Write a C++ code to perform insertion and deletion in queue using arrays applying the algorithms given in the manual. Create a menu where n is the number of disks.

2.2 Procedure: Task 1

```
#include<iostream>
using namespace std;

int main(){
    int a=-1,y=0;
    int array[100];
    char op;
    cout<<"To push enter p";
    cout<<"\nTo delete enter d";
```

```

cout<<"\nTo display enter s";
cout<<"\nTo exit enter e\n";
line:
cin>>op;
switch (op){

case 'p':
    cout<<"enter no to push\n";
    a++;
    cin>>array[a];
    cout<<"pushed at "<<a<<"\n";
    break;
case 'd':
    cout<<"dlting \n";
    y++;
    break;
case 's':
    for (int b=y;b<=a;b++){
        cout<<array[b]<<endl;
    }
    break;
case 'e':
    exit;
}
goto line;
}

```