G. H. Raisoni College Of Engineering And Management, Wagholi Pune

2021- 2022

Assignment no :- 4

Department	CE [SUMMER 2022 (Online)]		
Term / Section	III/B	Date Of **submission**	28-10-2021
Subject Name /Code	Data Structures and Algorithms/ UCSL201/UCSP201		
Roll No.	SCOB77	Name	Pratham Rajkumar pitty
Registration Number	2020AC0E1100107		

Experiment No.4

Aim :→ Queus are Frequently used in computer programming, and a typical example is the creation of a Job queue by an operating System. If the operating System does not use priorities, then the Jobs are processed in the order they enter the System.

- Write a C++ program for Simulating Job queue.
- Write Function to add Job and delete Job from queue.

Objective :→

To perform addition and deletion operation's on queue

- Input:

- Size of queue Elements in

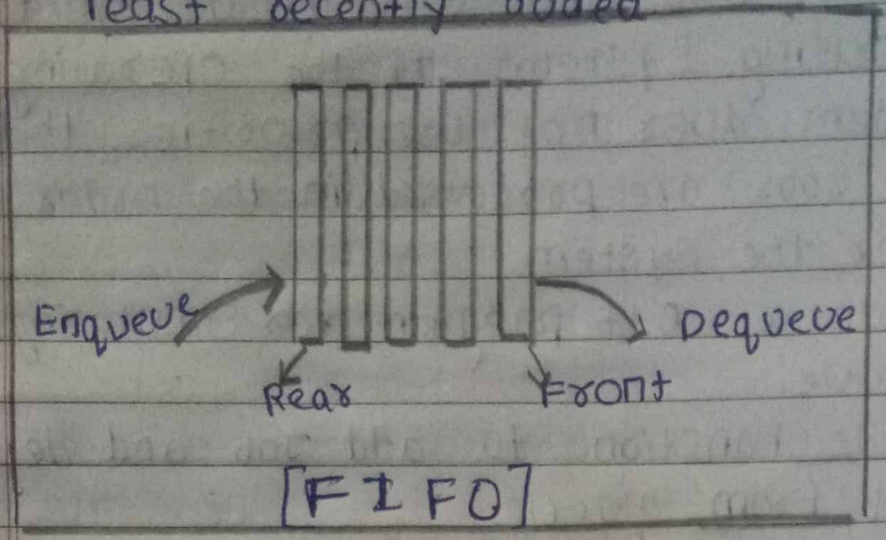
- output's / out come's:

- Result of addition of Job operation on ~~que~~ queue
- Result of deletion of Job operation on queue

Theory :→ • A queue is a ~~leo~~ linear Structure which Follows a particular order in which the operation are performed

- The order is "First In First Out" (FIFO)
- A good example of a queue is any queue of consumers For a resource where the consumer that come first is served First.
- The difference between Stack and queus is in Removing

- In a stack we remove the item the mos recently added;
  In a queue, we remove the item the least recently added



Applications of queue

- When a resourse is shared among multiple consumers. Ex. including CPU scheduling, Disk scheduling.
- When data is transferred asynchronously between two processes. Ex, IO. Buffers, pipes, File IO
- In Operating Systems:
  (a) Semaphores
  (b) FCFS (First come First serve) Scheduling
  Ex FIFO
  (c) Spooling in printers
  (d) Buffer for devices like keyboard
- In Networks:
  (a) queues in routers/ switches
  (b) Mail queus
- Variations: (
  (Deque, Priority queue, Doubly Ended priority Queue)

▶ Basic Operations

Basic operations associated with queues
- enqueue() - add (store) an item to the queue.
- dequeue() - remove (access) an item from the queue.

Few more operations are,
- peek() - Gets the element at the Front of the queue without removing it.
- isFull() - checks if the queue is Full.
- isempty() - checks if the queue is empty.

In queue, we always dequeue (or access) data, pointed by Front pointer and while enqueing (or storing) data in the queue we take help of rear pointer.

▶ Functions in queue

(1) peek() → Gets the element at the Front of the queue without Removing it.

Algorithm

```
begin procedure peek
return queue [Front]
end procedure
```

syntax Ex

```
int Peek() {
  return queue [Front];
}
```

(2) isFull() → check if the queue is Full.
while working on single dimentional

array to implement queue we just check For the rear pointer to rean at MAXSIZE to determine that the queue is Full.

Algorithm :→

```
begin procedure isfull
 if rear equals to MAXSIZE
     return true
 else
     return False
 endif
end procedure
```

Syntax:→

```
bool isfull () {
if (rear == MAXSIZE -1)
return true;
else
    return False ;
}
```

(3) isempty :→

Algorithm:→

```
begin procedure isempty
 if Front is less than MIN OR Front
                       is greater than
  return true;
 else
  return False;
 endif
 end procedure
```

Syntax :→

```
bool isempty () {
 if (Front <0 || Front > rear)
   return true;
 else
 return False; }
```

(4) Enqueue operation :→

queue maintains ~~mainly~~ 2 data pointers, Front and rear. Therefore, its operations are comparotively different to implement than that of stack.

Steps to Follow:→

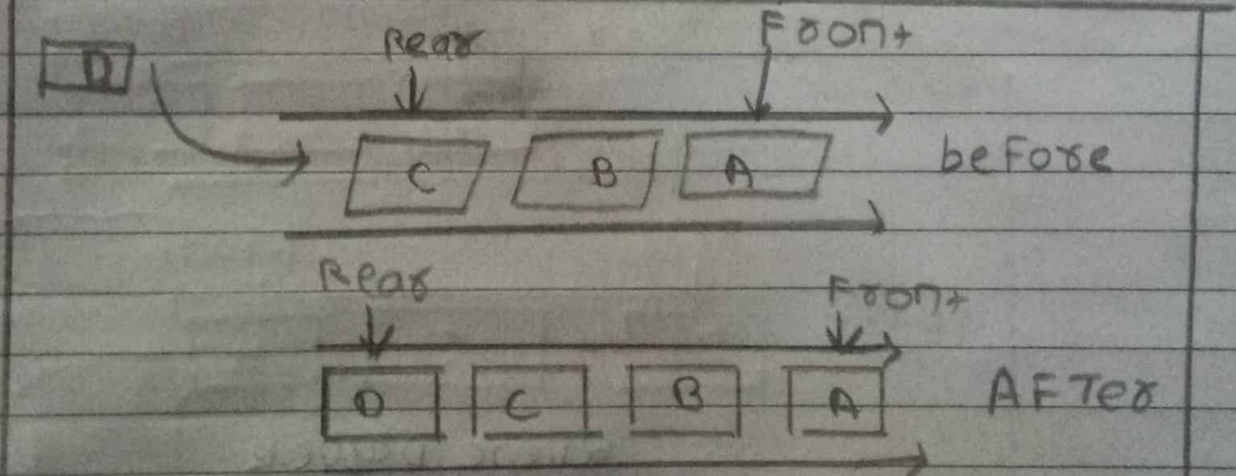
Step1 → check if the queue is full

Step2 → If the queue is Full, ~~provide~~ produce overfow error and exit.

Step3 → If the queue is not full, increment rear pointer to point the next empty space.

Step4 → Add data elements to the queue location where the rear is pointing

Step5 → return Success



queue Enqueue.

Algorithm:→

```
procedure enque (data)
 if queue is Full
   return overflow
endif
rear ← real +1
que [rear] ← data
return ~~toot~~true
end procedure
```

Syntax Ex.

```
int enqueue (int data)
if (isfull())
  return 0;
  rear = rear +1
  queue [rear] = data;
  return 1;
   end procedure
```

(5) Dequeue Operation:→

Accessing Data From queu is a 2 way process of two tasks - access the data where front is pointing and remov the data after access.

Steps to Follow:→

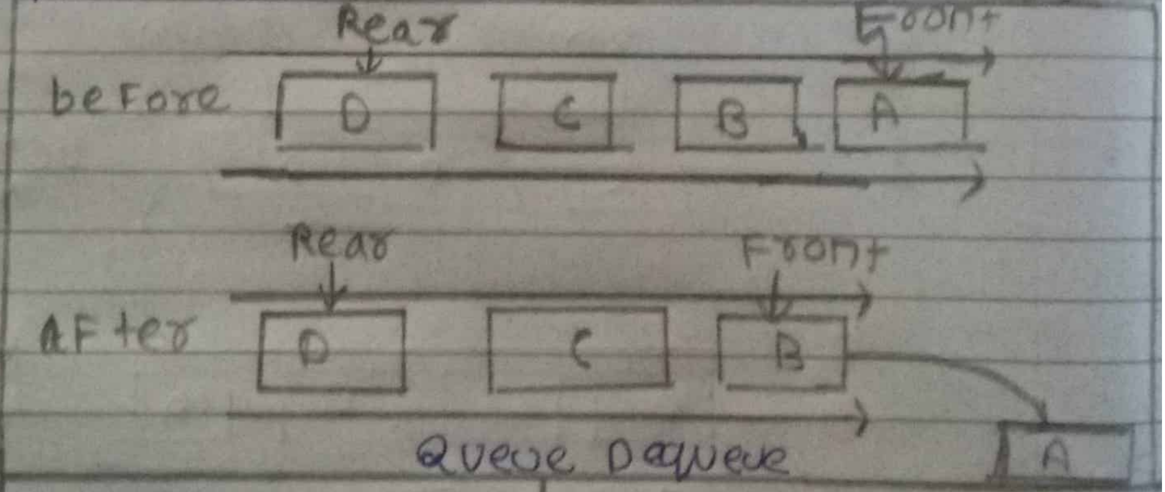
Step1:→ Check if the queue is empty

Step2:→ If the queue is empty perform underfl error and exit.

Step3:→ If the queue is not empty access th data where ~~queue~~ front is pointing

Step4:→ Increment front pointer to point to the next available data element.

Step5:→ Return success.



Algorithm :→

```
procedure dequeue
 if queue is empty
  return underflow
 endif
 data = queue [front]
 front = front+1
 return true
end procedure
```

Syntax Ex

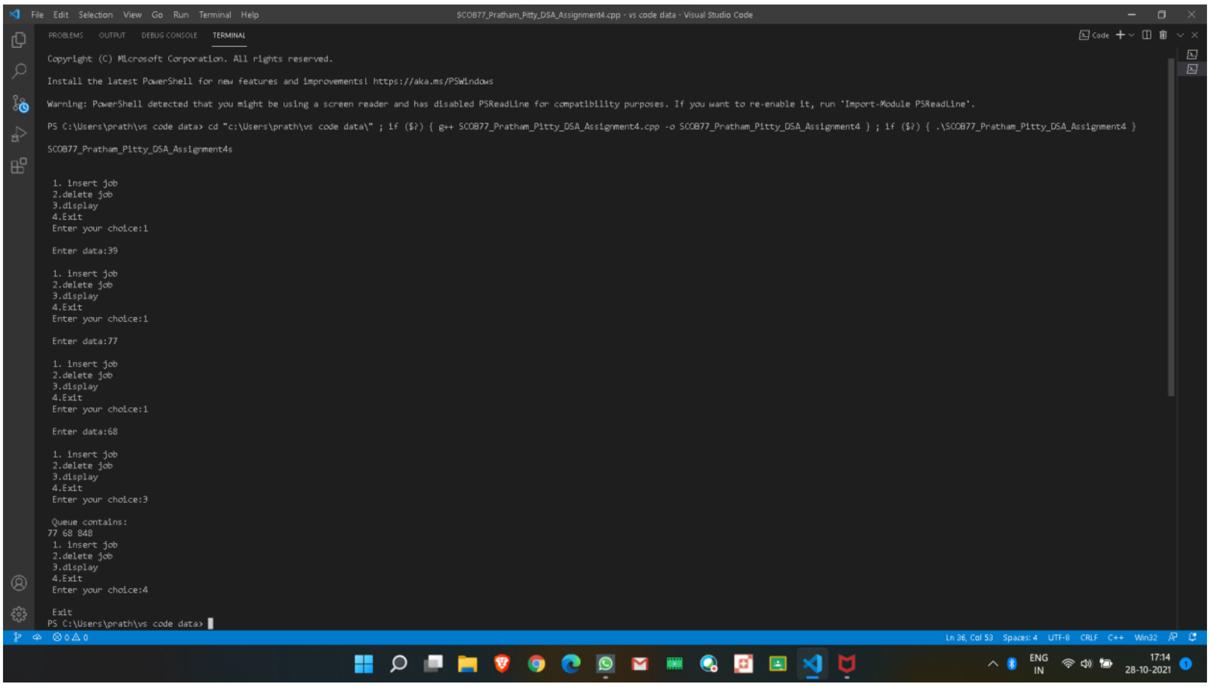
```
int dequeue() {
  if (isempty())
   return 0;
  int data = queue [fro
  front = front +1
  return data;
 }
```

Program code

```
#include <iostream>
#define MAX 10
using namespace std;
struct queue
{      int data[MAX];
            int front,rear;
};
class Queue
{    struct queue q;
   public:
      Queue(){q.front=q.rear=-1;}
      int isempty();
      int isfull();
      void enqueue(int);
      int delqueue();
      void display();
};
int Queue::isempty()
{
            return(q.front==q.rear)?1:0;
}
int Queue::isfull()
{    return(q.rear==MAX-1)?1:0;}
void Queue::enqueue(int x)
{q.data[++q.rear]=x;}
int Queue::delqueue()
```

```
{return q.data[++q.front];}
void Queue::display()
{   int i;
    cout<<"\n";
    for(i=q.front+1;i<=q.rear;i++)
                  cout<<q.data[i]<<" ";
}
int main()
{
      cout<<"\nSCOB77_Pratham_Pitty_DSA_Assignment4s\n\n";
     Queue obj;
             int ch,x;
             do{    cout<<"\n 1. insert job\n 2.delete job\n 3.display\n 4.Exit\n Enter your
choice:";
                    cin>>ch;
             switch(ch)
             {  case 1: if (!obj.isfull())
                             {   cout<<"\n Enter data:";
                                       cin>>x;
                                       obj.enqueue(x);
                             }
                       else
                                cout<< "Queue is overflow";
                        break;
                case 2: if(!obj.isempty())
                                           cout<<"\n Deleted Element="<<obj.delqueue();
                              else
                                       {   cout<<"\n Queue is underflow";  }
                              cout<<"\nremaining jobs :";
                              obj.display();
                  break;
              case 3: if (!obj.isempty())
                    {  cout<<"\n Queue contains:";
                            obj.display();
                    }
                    else
                              cout<<"\n Queue is empty";
                   break;
              case 4: cout<<"\n Exit";
         }
       }while(ch!=4);
return 0;
}
```

<u>Output:-</u>


File Edit Selection View Go Run Terminal Help
SCOB77_Pratham_Pitty_DSA_Assignment4.cpp - vs code data - Visual Studio Code
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
Ln 36, Col 53
Spaces: 4
UTF-8
CRLF
C++
Win32
ENG
IN
17:14
28-10-2021

```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\prath\vs code data> cd "c:\Users\prath\vs code data\" ; if ($?) { g++ SCOB77_Pratham_Pitty_DSA_Assignment4.cpp -o SCOB77_Pratham_Pitty_DSA_Assignment4 } ; if ($?) { .\SCOB77_Pratham_Pitty_DSA_Assignment4 }

SCOB77_Pratham_Pitty_DSA_Assignment4s


 1. insert job
 2.delete job
 3.display
 4.Exit
 Enter your choice:1

 Enter data:39

 1. insert job
 2.delete job
 3.display
 4.Exit
 Enter your choice:1

 Enter data:77

 1. insert job
 2.delete job
 3.display
 4.Exit
 Enter your choice:1

 Enter data:68

 1. insert job
 2.delete job
 3.display
 4.Exit
 Enter your choice:3

 Queue contains:
77 68 848
 1. insert job
 2.delete job
 3.display
 4.Exit
 Enter your choice:4

 Exit
PS C:\Users\prath\vs code data> 
```