Experiment2.4

Student Name: Rohit Kumar Mahato **UID: 21BCS7480**
Branch: CSE **Section/Group: 717A**
Semester: 3 **Date of Performance: 27/10/2022**
Subject Name: Data Structure **Subject Code:21CSH-211**

Aim:- Write a program to demonstrate the implementation of various operations on a linear queue and circular represented using a linear array.

Algorithm:-

For linear queue: insertion

1. Check the base case, rear=max-1, print overflow and exit.
2. If FRONT=REAR=-1
3. SET FRONT =REAR=0
4. Else REAR=REAR+1
5. SET QUEUE(REAR)=item
6. Exit

For linear queue: deletion

1. If front =-1 || front>rear then underflow and exit
2. Else set val=queue(front)
3. front =front+1
4. exit

For circular linear queue: insertion

1. if (rear+1) % max = front then overflow and exit
2. if front= rear = -1 set front=rear = 0
3. else if rear =max – 1 and front !=0
4. set rear =0
5. else rear =(rear+1) % max
6. set queue(rear) = value and exit

For circular linear queue: deletion

1. if front=-1 underflow and exit
2. set val=queue(front)

3. if front = rear
4. set front = rear – 1 else
5. if front = max-1
6. set front= 0
7. else set front =front+1
8. exit

Linear Queue Code:-

```
#include <iostream>
using namespace std;
int queue[10], n = 10, front = - 1, rear = - 1;
void Insert() {
   int val;
   if (rear == n - 1)
   cout<<"Queue Overflow"<<endl;
   else {
      if (front == - 1)
      front = 0;
      cout<<"Insert the element in queue : "<<endl;
      cin>>val;
      rear++;
      queue[rear] = val;
   }
}
void Delete() {
   if (front == - 1 || front > rear) {
      cout<<"Queue Underflow ";
      return ;
   } else {
      cout<<"Element deleted from queue is : "<< queue[front] <<endl;
      front++;;
   }
}
void Display() {
   if (front == - 1)
   cout<<"Queue is empty"<<endl;
   else {
      cout<<"Queue elements are : ";
      for (int i = front; i <= rear; i++)
      cout<<queue[i]<<" ";
         cout<<endl;
   }
}
int main() {
   int ch;
   cout<<"1) Insert element to queue"<<endl;
```

```
    cout<<"2) Delete element from queue"<<endl;
    cout<<"3) Display all the elements of queue"<<endl;
    cout<<"4) Exit"<<endl;
    do {
        cout<<"Enter your choice : "<<endl;
        cin>>ch;
        switch (ch) {
            case 1:
            Insert();
            break;
            case 2:
            Delete();
            break;
            case 3:
            Display();
            break;
            case 4:
            cout<<"Sub Khatam hoo gaya"<<endl;
            break;
            default:
            cout<<"Invalid choice"<<endl;
        }
    } while(ch!=4);
    return 0;
}
```

Output:-

```
PS D:\desktop\vscode> cd "d:\desktop\vscode\Queues\" ; if ($?) { g++ Inser
1) Insert element to queue
2) Delete element from queue
3) Display all the elements of queue
4) Exit
Enter your choice :
1
Insert the element in queue :
2
Enter your choice :
1
Insert the element in queue :
5
Enter your choice :
1
Insert the element in queue :
6
Enter your choice :
1
Insert the element in queue :
7
Enter your choice :
1
Insert the element in queue :
8
Enter your choice :
3
Queue elements are : 2 5 6 7 8
Enter your choice :
2
Element deleted from queue is : 2
```

Circular Queue Code:-

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

int cqueue[5];
int front = -1, rear = -1, n=5;

void insertCQ(int val) {
   if ((front == 0 && rear == n-1) || (front == rear+1)) {
      cout<<"Queue Overflow ";

      return;
   }
   if (front == -1) {
      front = 0;
      rear = 0;
   } else {
      if (rear == n - 1)
      rear = 0;
      else
      rear = rear + 1;
   }
   cqueue[rear] = val ;
}
void deleteCQ() {
   if (front == -1) {
      cout<<"Queue Underflow";
      return ;
   }
   cout<<"Element deleted from queue is : "<<cqueue[front]<<endl;

   if (front == rear) {
      front = -1;
      rear = -1;
   } else {
      if (front == n - 1)
      front = 0;
      else
      front = front + 1;
   }
}
void displayCQ() {
   int f = front, r = rear;
   if (front == -1) {
      cout<<"Queue is empty"<<endl;
      return;
   }
   cout<<"Queue elements are :";
```

```
    if (f <= r) {
        while (f <= r){
            cout<<cqueue[f]<<" ";
            f++;
        }
    } else {
        while (f <= n - 1) {
            cout<<cqueue[f]<<" ";
            f++;
        }
        f = 0;
        while (f <= r) {
            cout<<cqueue[f]<<" ";
            f++;
        }
    }
    cout<<endl;
}
int main() {

    int ch, val;
    cout<<"1)Insert"<<endl;
    cout<<"2)Delete"<<endl;
    cout<<"3)Display"<<endl;
    cout<<"4)Exit";
    do {
        cout<<"Enter choice : "<<endl;
        cin>>ch;
        switch(ch) {
            case 1:
            cout<<"Input for insertion: "<<endl;
            cin>>val;
            insertCQ(val);
            break;
            case 2:
            deleteCQ();
            break;
            case 3:
            displayCQ();
            break;
            case 4:
            cout<<"Exit";
            break;
            default: cout<<"Incorrect!";
        }
    } while(ch != 4);
    return 0;
}
```

Output

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

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

PS D:\desktop\vscode> cd "d:\desktop\vscode\Queues\" ; if ($?) { g++ circular_queue.cpp
1)Insert
2)Delete
3)Display
4)ExitEnter choice :
1
Input for insertion:
4
Enter choice :
1
Input for insertion:
5
Enter choice :
1
Input for insertion:
8
Enter choice :
1
Input for insertion:
2
Enter choice :
3
Queue elements are :4 5 8 2
Enter choice :
2
Element deleted from queue is : 4
Enter choice :
```

Learning Outcomes (What I have learned)

1. Learnt about queue and its types
2. Also learnt the various operations on queue