EXPERIMENT – 2.4

Name: Rohan Jaiswal
UID: 21BCS2856
Branch: CSE
Section/Group: 608 (B)
Semester: 3rd
Date of Performance: 13th Oct
Subject Name: DS
Subject Code: 21CSH-211

Aim of Practical:

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

Algorithm:

1. If rear=max
   print overflow
   Exit;
2. If front=rear=-1
   then set front=rear=0;
   else
   rear++;
3. Queue[rear] = item
4. Exit

Program Code:

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

struct Queue{
    int front, rear, size;
    int* arr;

    Queue(int s){
        front=rear=0;
        size=s;
        arr = new int[size];
    }

    void display(){
        if(rear==front){
            cout<<"Queue is Empty!\nFirst Insert some elements :)\n";
            return;
        }
        cout<<"Queue Elements: ";
        for(int i=0;i<rear;++i)
            cout<<arr[i]<<" ";
        cout<<"\n";
    }

    void getFront(){
        if(rear==front)
            cout<<"Queue is Empty!\n";
        else{
            cout<<"First Element: "<<arr[0]<<"\n";
        }
    }

    void push(int val){
        if(rear>=size){
            cout<<"Insertion Unsuccesful!\nQueue is Full.\n";
        }
        else{
            arr[rear]=val;
            rear++;
        }
```

```
    }

    void pop(){
        if(rear==front){
            cout<<"Deletion Unsuccesful!\nQueue is Empty!\n";
        }
        else{
            for(int i=0;i<rear-1;++i)
                arr[i]=arr[i+1];
            rear--;
        }
    }
};

int main(){
    cout<<"Enter Size of the Queue: ";
    int size;cin>>size;

    Queue q(size);

    bool flag = true;
    while(flag){
        std::cout<<"\nQueue basic operation Menu :-\n";
        std::cout<<"1. Display\n2. Get Front\n3. push.\n4. Delete.\n5. Exit
Program\n\n";
        std::cout<<"Your choice: ";
        std::string choice; std::cin>>choice;

        if(choice.size()>1) // for tackling when input is alphabet and strings.
            choice[0]='6';

        switch (choice[0])
        {
        case '1':
            q.display();
            break;
        case '2':
            q.getFront();
            break;
        case '3':
            std::cout<<"Enter the value of Element you want to push: ";
            int val;std::cin>>val;
```

```
            q.push(val);
            break;
        case '4':
            q.pop();
            break;
        case '5':
            flag = false;
            std::cout<<"Exiting.......";
            break;
        default:
            std::cout<<"Invalid Choice... try again!";
            break;
        }
        std::cout<<"\n";
        system("pause");
        std::cout << "\033[2J\033[1;1H"; //for clearing screen in terminal.
    }
    std::cout<<"Program Stopped!!";
}
```

Output:

```
Enter Size of the Queue: 2

Queue basic operation Menu :-
1. Display
2. Get Front
3. push.
4. Delete.
5. Exit Program

Your choice: 3
Enter the value of Element you want to push: 5

Press any key to continue . . .
```

```
Queue basic operation Menu :-
1. Display
2. Get Front
3. push.
4. Delete.
5. Exit Program

Your choice: 3
Enter the value of Element you want to push: 7

Press any key to continue . . .
```

```
Queue basic operation Menu :-
1. Display
2. Get Front
3. push.
4. Delete.
5. Exit Program

Your choice: 1
Queue Elements: 5 7

Press any key to continue . . .
```

```
Queue basic operation Menu :-
1. Display
2. Get Front
3. push.
4. Delete.
5. Exit Program

Your choice: 2
First Element: 5

Press any key to continue . . .
```