**PRACTICAL ASSIGNMENT::3**

**GITHUB:https://github.com/Atharva-byte1/DSA**

**Implement Circular Queue using Array. Perform following operations on it.**

**a) Insertion (Enqueue)**

**b) Deletion (Dequeue)**

**c) Display**

**(Note: Handle queue full condition by considering a fixed size of a queue.)**

#include <bits/stdc++.h>

#define MAXSIZE 3

using namespace std;

int q[MAXSIZE],r=-1,f=-1;

void display() {

if (f == -1) {

cout << "\nQueue is empty\n";

return;

}

cout << "\n\n\t" << "Element" << "\t\t" << "Position";

for (int i = f; i <= r; i++) {

cout << "\n\n\t" << q[i] << "\t\t" << i;

}

cout << endl;

}

int insert(){

int ele;

if(f==(r+1)%MAXSIZE){

cout<<"QUEUE is full \n";

}

else{

if (f == -1) {

f = 0;

}

r=(r+1)%MAXSIZE;

cout<<"enter the number \n";

cin>>ele;

q[r]=ele;

cout<<ele<<"is inserted at"<<r<<endl<<endl;

}

}

int del(){

int ele;

if(f==-1||f>r){

cout<<"Q is empty\n";

}

else{

q[f]=ele;

cout<<"\n the number is deleted"<<ele;

f=(f+1)%MAXSIZE;

if(r==(MAXSIZE-1)&&f==0){

f=r=-1;

}

}

}

int main(){

int ch;

do{

cout<<"1.insert the number\n";

cout<<"2 delete the number\n";

cout<<"3.display the output\n";

cout<<"4.exit\n";

cin>>ch;

switch(ch){

case 1:

insert();

break;

case 2:

del();

break;

case 3:

display();

break;

case 4:

cout<<"exit";

break;

default:

cout<<"!!!INVALID INPUT";

}

} while(ch!=4);

return 0;

}

