**[//Implementation](\\\\Implementation) of circular queue**

#include<iostream>

#define max 5

using namespace std;

class queue

{

int a[max];

int front, rear;

public :

queue()

{

front=-1;

rear=-1;

}

int isempty()

{

if(rear==-1 || front==-1)

return 1;

else

return 0;

}

int isfull()

{

if((rear+1)%max==front) //changed.....

return 1;

else

return 0;

}

void enqueue(int value); //function declare

void dequeue();

void display();

};

int main()

{

int value, choice;

queue q;

do

{

cout<<"Select any one operation: 1. Enqueue 2. Dequeue 3. Display 4. Exit\n";

cin>>choice;

switch(choice)

{

case 1: cout<<"Enter value to be inserted into queue\n" ;

cin>>value;

q.enqueue(value); //function call

break;

case 2:q.dequeue();

break;

case 3:q.display();

break;

case 4:cout<<"Exit\n";

break;

default: cout<<"Wrong choice\n";

break;

}

}while(choice !=4);

}

void queue :: enqueue(int value)

{

int x;

x=isfull();

if(x==1)

cout<<"Queue is already full..cannot insert more elements\n";

else

if(front==-1 && rear==-1)

{

front=0;

rear=0;

}

else

if(rear==max-1 && front!=0)

rear=0;

else

{

rear=(rear+1)%max; //changed....

}

a[rear]=value;

}

void queue :: dequeue()

{

int x;

x=isempty(); //calling isempty() function

if(x==1)

cout<<"Queue is empty...Cannot delete element\n";

else

if(front==rear)

{

cout<<"Deleted element is="<<a[front]<<"\n";

front=-1;

rear=-1;

}

else

if(front == max -1)

{

cout<<"Deleted element is="<<a[front]<<"\n";

front = 0;

}

else

{

cout<<"Deleted element is="<<a[front]<<"\n";

front++; //changed...

}

}

void queue :: display()

{

int i;

if(front==-1)

cout<<"Queue is empty\n";

else

{

i=front;

while(i!=rear)

{

cout<<a[i]<<"\t";

i=(i+1)%max;

}

}

}