**ALGORITHM:**

**Enqueue**

1. Start
2. If the stack is empty,then read the data
3. Increment the rear and store the data in the rear index of q
4. Stop

**Dequeue**

1. Start
2. Set the element to be deleted as q[front]
3. Increment front
4. Stop

**Is empty**

1. Start
2. If rear is less than front then return 1
3. Otherwise return 0
4. Stop

**Is full**

1. If rear is equal to one less than the size then return 1
2. Otherwise return 0
3. Stop

**Display**

1. Start
2. From front as the first index and rear as the last index print all the elements
3. Stop

**PROGRAM:**

#include<iostream>

using namespace std;

class queue

{

public:

int q[5];

int front=0;

int rear=-1;

void enqueue()

{

int data;

int l=isfull();

if(!l)

{

cout<<"\nEnter the data\t";

cin>>data;

rear++;

q[rear]=data;

}

}

void dequeue()

{

if(!isempty())

{

int deleter;

deleter=q[front];

front ++;

}

}

int isempty()

{

if(rear<front)

{

return 1;

}

else

{

return 0;

}

}

int isfull()

{

if(rear==4)

{

return 1;

}

else

{

return 0;

}

}

void display()

{

for(int i=front;i<=rear;i++)

{

cout<<q[i];

}

}

};

main()

{

queue a;

int ch,op;

do

{

cout<<"\nEnter choice";

cout<<"\nMenu...\n1.Insert\n2.Delete\n3.Display\n";

cin>>ch;

switch(ch)

{

case 1:

{

a.enqueue();break;

}

case 2:

{

a.dequeue();

cout<<"\nDisplaying....\n";

a.display();

break;

}

case 3:

{

cout<<"\nDisplaying...\n";

a.display();

break;

}

}

cout<<"\nDo you want to continue(0/1)";

cin>>op;

}while(op==1);

}

**SAMPLE INPUT AND OUTPUT:**

Enter choice

Menu...

1.Insert

2.Delete

3.Display

1

Enter the data 2

Do you want to continue (0/1) 1

Enter choice

Menu...

1.Insert

2.Delete

3.Display

1

Enter the data 3

Do you want to continue (0/1) 1

Enter choice

Menu...

1.Insert

2.Delete

3.Display

3

Displaying...

2 3

Do you want to continue (0/1 )1

Enter choice

Menu...

1.Insert

2.Delete

3.Display

2

Displaying....

3

Do you want to continue (0/1) 0