Ex. No. 1b **Queue Array**

Date:

**Aim:**

To implement queue operations using array.

**Algorithm:**

1. Start

2. Define a array queue of size max = 5

3. Initialize front = rear = –1

4. Display a menu listing queue operations

5. Accept choice

6. If choice = 1 then

If rear < max -1

Increment rear Store element at current position of rear

Else

Print Queue Full

Else If choice = 2 then

If front = –1 then

Print Queue empty

Else

Display current front element

Increment front

Else If choice = 3 then

Display queue elements starting from front to rear.

7. Stop

**Program**

#include<stdio.h>

#include<stdlib.h>

#define SIZE 5

int front = - 1;

int rear = - 1;

int q[SIZE];

void insert( );

void del( );

void display( );

void main( )

{

int choice;

do

{

printf("\n\t Menu");

printf("\n 1. Insert");

printf("\n 2. Delete");

printf("\n 3. Display ");

printf("\n 4. Exit");

printf("\n Enter Your Choice:");

scanf("%d", &choice);

switch(choice)

{

case 1:

insert( );

display( );

break;

case 2:

del( );

display( );

break;

case 3:

display();

break;

case 4:

printf("End of Program....!!!!\n");

exit(0);

}

}while(choice != 4);

}

void insert( )

{

int no;

printf("\n Enter No.:");

scanf("%d", &no);

if(rear < SIZE - 1)

{

q[++rear]=no;

if(front == -1)

front=0;// front=front+1;

}

else

{

printf("\n Queue overflow");

}

}

void del( )

{

if(front == - 1)

{

printf("\n Queue Underflow");

return;

}

else

{

printf("\n Deleted Item:-->%d\n", q[front]);

}

if(front == rear)

{

front = - 1;

rear = - 1;

}

else

{

front = front + 1;

}

}

void display( )

{

int i;

if( front == - 1)

{

printf("\nQueue is empty....");

return;

}

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

printf("\t%d",q[i]);

}

**Output:**

**Result**

Thus insert and delete operations of a queue was demonstrated using arrays.