**Experiment No. 4**

**Title :** Implementation of Queue Linear Data Structure

**Problem Statement :** Implementing linear data structure queue using array with functions

Enqueue()

Dequeue()

Display()

**Algorithm :**

**S1 :** Start

**S2 :** Declare an array along with the size, front and rear variables to keep a track of the index of stack, a choice variable

**S3 :** Call the functions in a switch statement according to the choice value

**S4 :** In enqueue function check if front=-1 if so make front value to 0 and increment the rear value and with that as index put the value in queue array

**S5 :** In dequeue function check if front value is -1 or greater than rear value if so queue is empty else remove the element at index equal to front.

**S6 :** In display function we display the queue from front to rear.

**S7 :** Stop

**Code :**

#include<stdio.h>

# define MAX 5

int front = -1;

int rear = -1;

int queue[];

void enqueue()

{

int var;

if(front==-1)

front = 0;

// if(rear >= MAX - 1)

// {

// printf("Queue is FULL\n");

// return;

// }

printf("Enter the value :");

scanf("%d",&var);

rear = rear + 1;

queue[rear] = var;

printf("%d is queued\n",queue[rear]);

}

void dequeue()

{

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

{

printf("Queue is EMPTY\n");

return;

}

printf("%d is dequeued\n",queue[front]);

front = front + 1;

}

void display()

{

int i;

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

{

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

}

printf("\n");

}

int main()

{

int ch;

start :

printf("Enter the choice\n1.Enqueue\t2.Dequeue\t3.Display\t");

scanf("%d",&ch);

switch(ch)

{

case 1 :

enqueue();

break;

case 2 :

dequeue();

break;

case 3 :

display();

break;

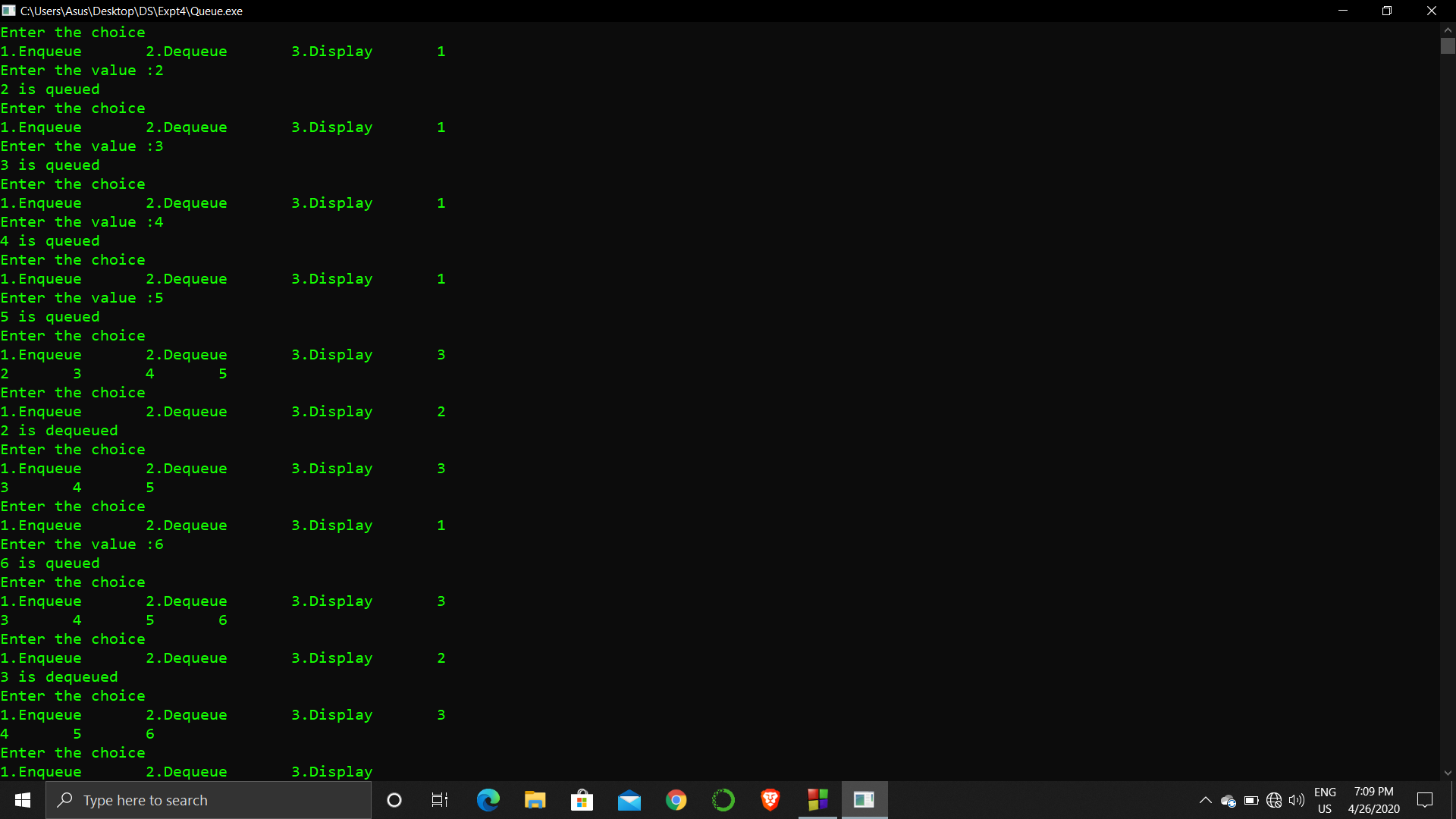
default :

printf("Enter the correct choice\n");

}

goto start

**Output :**

****

**Analysis :**

The queue size is limited which makes the operations related to it also limited.

If queue is once full and if we dequeue all the values and then try to enqueue again it leads to an error since queue size if limited.

If the front and rear value gets compromised the queue gives and error since they are the only values that control the operation of queue.