**Experiment No. 4**

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

**Problem Statement :** Write a C code to implement Queue data structure with functionalities of Enqueue(), Dequeue(),Display()

**Algorithm :**

**Step 1:** Start

**Step 2:** Declare a stack array and also front and back variables to keep track of the values of queue

**Step 3:** Declare Enqueue() function to insert an element into the queue from the rear end in which check if front is -1 if yes then make it 0 and increment rear and put new value in queue at rear index

**Step 4:** Declare Dequeue function to remove a value from the front end of the queue in which check front value is -1 or greater than rear then queue is empty else remove item from queue

**Step 5:** Declare a display function to display the queue from front to rear index

**Step 6:** Input the user choice for above three functions and switch to that function to operate on queue

**Step 7:** Stop

**Program:**

#include<stdio.h>

# define MAX 5

int front = -1;//to track front position of queue

int rear = -1;//to track rear position of queue

int queue[MAX]; //Declare Queue of type int

void enqueue()//enqueue function to insert value in queue

{

int val;

if(front==-1)//make index correction for array indexing

front = 0;

if(rear >= MAX - 1)//check if queue is full

{

printf("Queue is FULL\n");

return;

}

printf("Enter the value :");

scanf("%d",&val);//input value to enqueue

rear = rear + 1;//increment rear as inserting a value

queue[rear] = val;//insert value into queue

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

}

void dequeue() // dequeue function to remove value from queue from front

{

if(front== -1 || front > rear)//check if queue is empty

{

printf("Queue is EMPTY\n");

return;

}

printf("%d is dequeued\n",queue[front]);//dequeue element at the front

front = front + 1;//update front by incrementing

}

void display() // display function to display queue

{

int i;

printf("\nQUEUE is \n");

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

{

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

}

printf("\n");

}

int main()

{

int choice;//to store choice of user

while(1)

{

printf("Enter the choice\n1\_Enqueue\t2\_Dequeue\t3\_Display\t4\_exit\n");

scanf("%d",&choice); //input choice

switch(choice)

{

case 1 ://insert value

enqueue();

break;

case 2 ://take out value from front

dequeue();

break;

case 3 ://display queue

display();

break;

case 4:exit(0);break;

default :

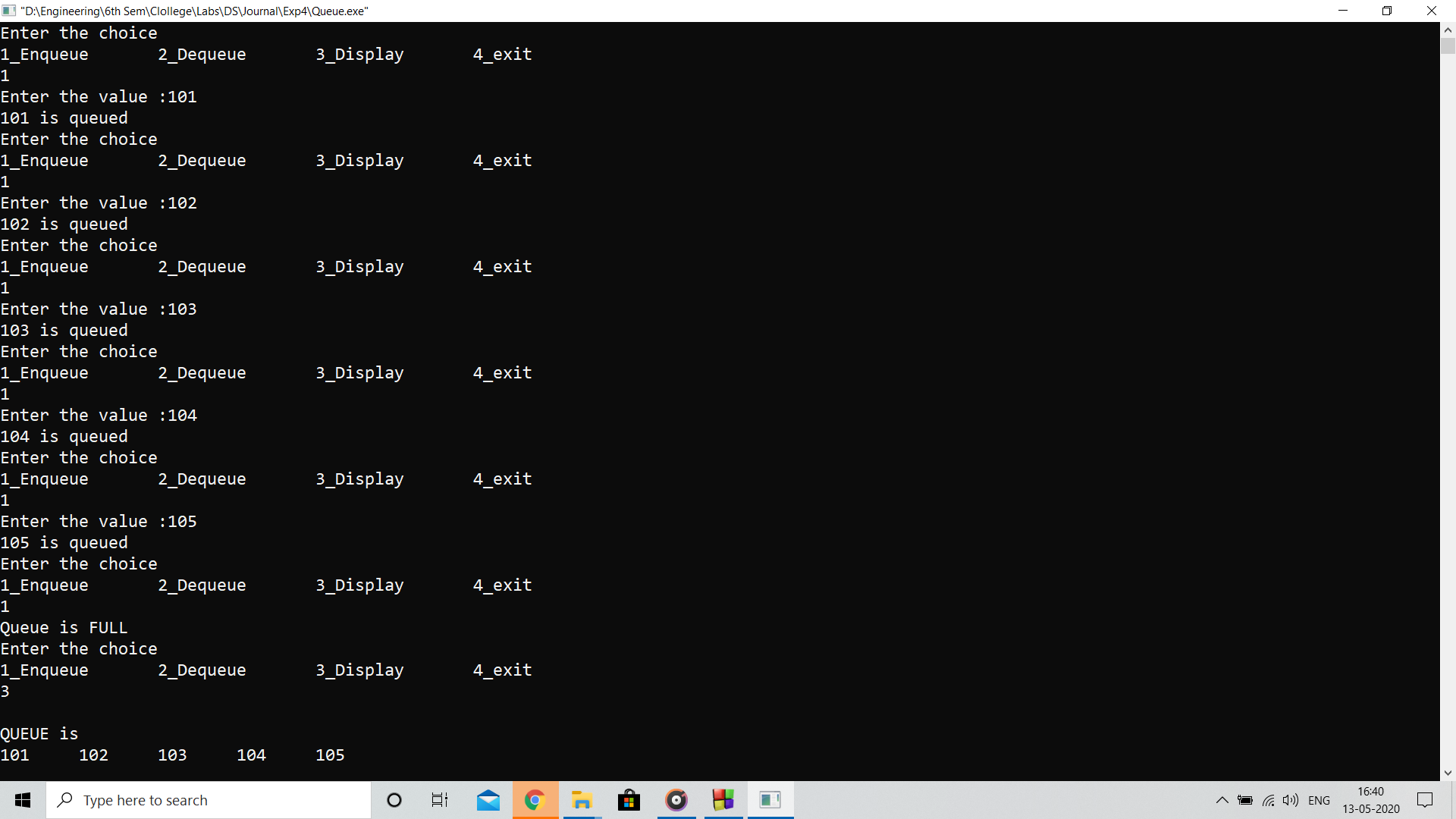
printf("Enter the correct choice\n");

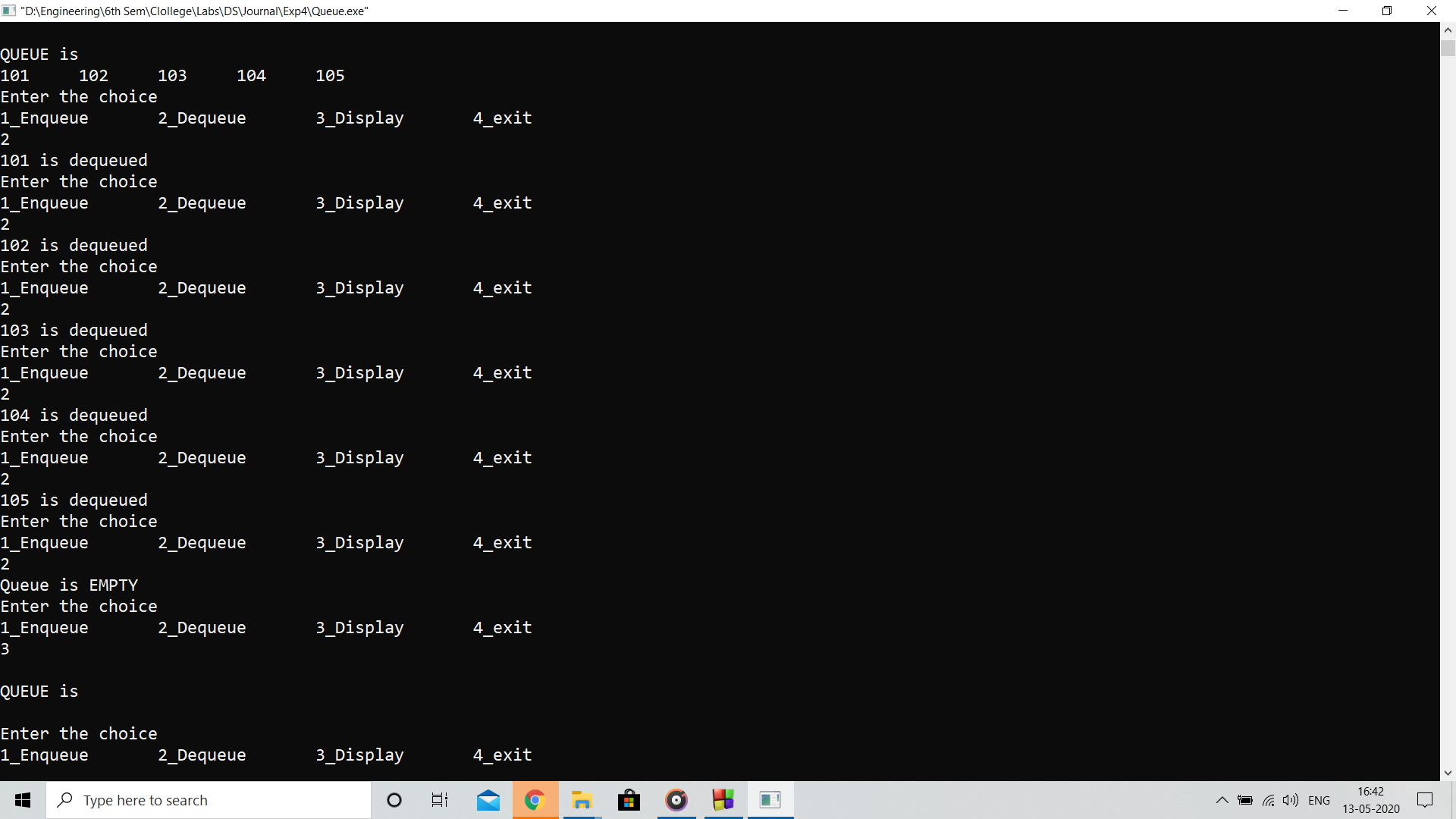
}//end switch

}//end while

}//end main

**Output:**

****

****

**Analysis:**

Program performs queuing operation and de-queuing operation with the use of front and rear variable for index reference

**Limitations:**

* Program fails to enqueue after the complete occupancy of values in the queue although you dequeue elements
* Queue size limits the operation of the queue which can be improved using dynamic memory allocation