

**Instructions: Please read carefully**

- Please rename this file as only your ID number (e.g. 18-\*\*\*\*\*-1.doc or 18-\*\*\*\*\*-1.pdf).
- Submit the file before **11:59pm on 12/03/2021** in the Portal Midterm Assignment section labeled Lab Assignment.

**Do not copy from others! !**

**Name:- Amit Podder**

**ID:- 20-42273-1**

**Section:- [ F ]**

**1. (a) Define STACK and its operation PUSH and POP for the following scenario. STACK maxsize = 6. Illustrate the following steps. (You should use pen and paper for answering this portion. Take a picture of the page and insert in answer section)**

Push(a);  
Push(b);  
Push(c);  
Pop();  
Push(d);  
Pop();  
Push(e);  
Push(f);  
Pop();  
Pop();  
Pop();

**Answer:-**

## Data Structure (Lab)

### Assignment

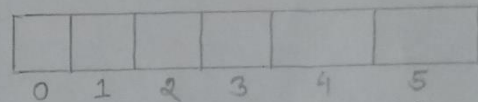
#### Ans to the ques No:- 1

A Stack is an abstract datatype that serves as a collection of elements. It has two main operations. One is Push and another is Pop. In Stack, the objects are inserted and removed according to LIFO principle.

Now, we are going to show the scenario of Push and Pop

Operation :-

Max size = 6 →



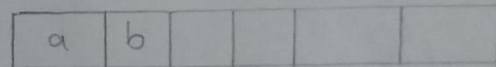
Push (a);

Push (b);

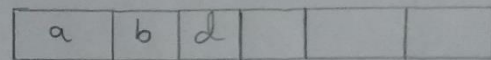
Push (c); →



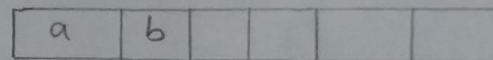
Pop (); →



Push (d); →

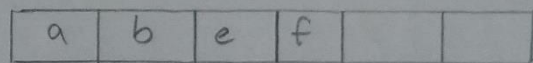


Pop (); →



Push (e);

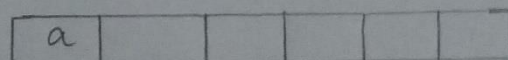
Push (f); →



Pop ();

Pop ();

Pop (); →



**Answer:-**

```
#include <iostream>
#define MaxSize 6

char a[MaxSize];
char top=-1;

void Push(int x)
{
    if(top==MaxSize-1)
    {
        std::cout<<"Stack Overflowed!!!"<<std::endl;
        return;
    }
    a[++top]=x;
}

void Pop()
{
    if(top== -1)
    {
        std::cout<<"Stack is Empty"<<std::endl;
        return;
    }
    top--;
}

int Top()
{
    return a[top];
}

void Print()
{
    if(top>=0)
    {
        int i;
        std::cout<<"Stack"<<std::endl;
        for(i=top;i>=0;i--)
        {
            std::cout<<" "<<a[i];
            std::cout<<std::endl;
        }
    }
}

using namespace std;
int main()
{
    Push('a');

    Print();
}
```

```
Push('b');  
  
Print();  
  
Push('c');  
  
Print();  
  
Pop();  
  
Print();  
  
Push('d');  
  
Print();  
  
Pop();  
  
Print();  
  
Push('e');  
  
Print();  
  
Push('f');  
  
Print();  
  
Pop();  
  
Print();  
  
Pop();  
  
Print();  
  
Pop();  
  
Print();  
  
return 0;  
}
```

**1. (c) Your example code's Console Output Screenshot here:-  
(insert Console Output Screenshot)**

**Answer:-**

```
C:\Users\USER\Desktop\1\bin\Debug\1.exe
Stack
a
Stack
b
a
Stack
c
b
a
Stack
b
a
Stack
d
b
a
Stack
b
a
Stack
e
b
a
Stack
f
e
b
a
Stack
e

C:\Users\USER\Desktop\1\bin\Debug\1.exe
Stack
d
b
a
Stack
b
a
Stack
e
b
a
Stack
f
e
b
a
Stack
e
b
a
Stack
b
a
Stack
a

Process returned 0 (0x0)   execution time : 0.068 s
Press any key to continue.
```

**2. (a) Define a Circular QUEUE and its operation for maxsize = 4. Illustrate the following steps. (You should use pen and paper for answering this portion. Take a picture of the page and insert in answer section)**

```
enqueue(a);
enqueue (b);
enqueue (c);
dequeue();
dequeue ();
enqueue (d);
dequeue ();
enqueue (e);
```

```
enqueue (f);  
dequeue ();  
dequeue ();
```

**Answer:-**

## Ans to the ques No:- 2

Circular Queue is a linear data structure in which the operations are performed based on FIFO principle and the last position is connected back to the first position to make a circle.

Now, we are going to show the scenario of Enqueue and Dequeue

Operation :-

enqueue (a);

enqueue (b);

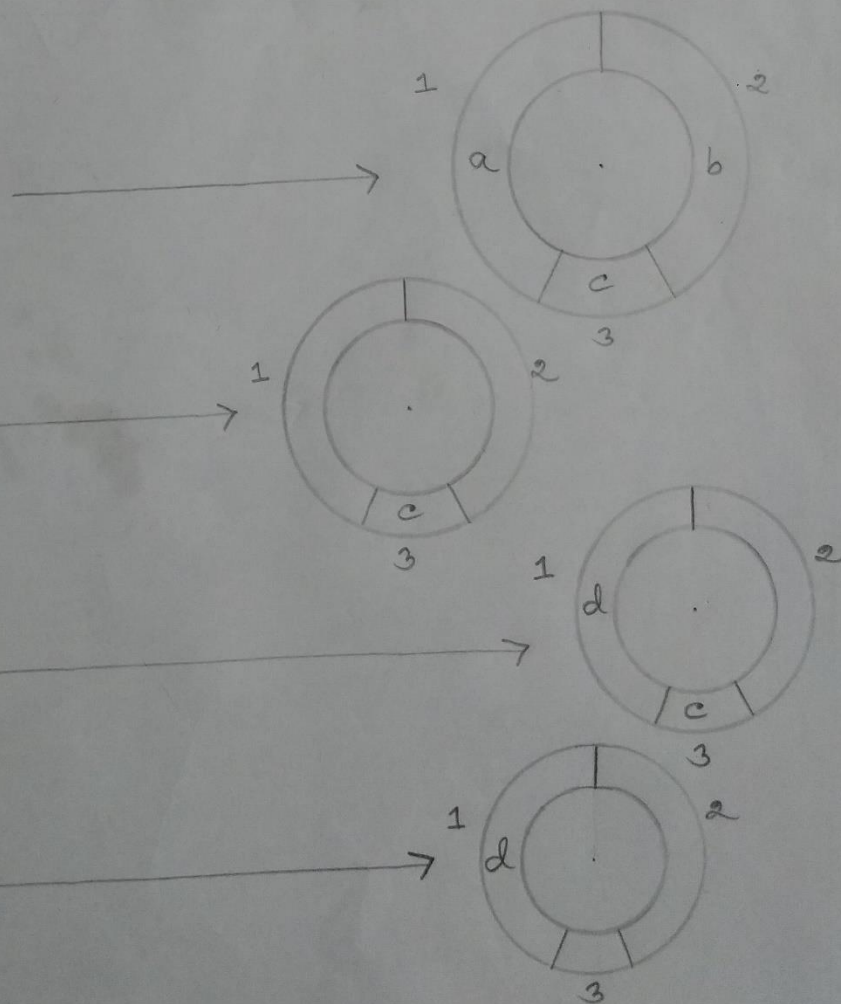
enqueue (c);

dequeue ();

dequeue ();

enqueue (d);

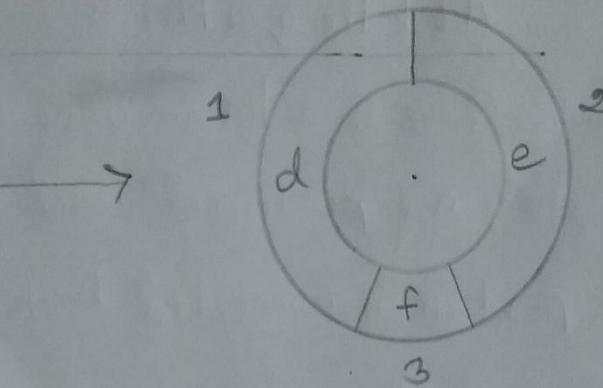
dequeue ();





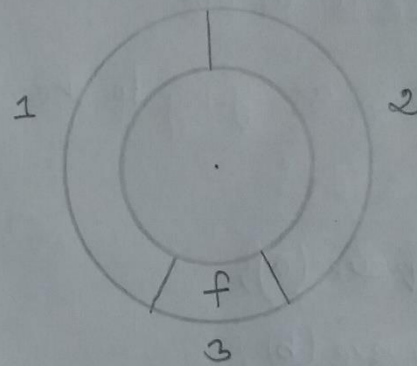
enqueue(e);

enqueue(f);



dequeue();

dequeue();



2. (b) Your example code here (Comment in the code):-

Answer:-



```
#include <iostream>
using namespace std;
#define MaxSize 4

//Circular Queue

char Queue[MaxSize];
char front=-1;
char rear=-1;

void enQueue(int item)
{
    if((front==0 && rear==MaxSize-1) || (front==rear+1))
    {
        cout<<"Queue Overflowed!!!"<<endl;
    }
    if(front==-1 && rear==-1)
    {
        front=0;
        rear=0;
    }
    else if(rear==MaxSize-1)
    {
        rear=0;
    }
    else{
        rear=rear+1;
        Queue[rear]=item;
    }
}

void deQueue()
{
    if(front==-1 && rear==-1)
    {
        cout<<"Queue Underflowed!!!"<<endl;
    }
    int item=Queue[front];
    if(front==rear && front==-1 && rear==-1)
    {
        cout<<"Queue Underflowed!!!"<<endl;
    }
    if(front==rear)
    {
        front=-1;
        rear=-1;
    }
    else if(front==MaxSize-1)
    {
        front=0;
    }
}
```

```
else
{
    front=front+1;
}
}

void show()
{
    if(rear==-1 && front==-1)
    {
        cout<<"Queue Underflowed!!!"<<endl;
    }
    else
    {
        int item;
        cout<<"Queue"<<endl;
        for(int i=front;i<=rear;i++)
        {
            cout<<Queue[i]<<endl;
        }
    }
}

int main()
{
    enQueue('a');

    enQueue('b');

    enQueue('c');

    deQueue();

    deQueue();

    enQueue('d');

    deQueue();

    enQueue('e');

    enQueue('f');

    deQueue();

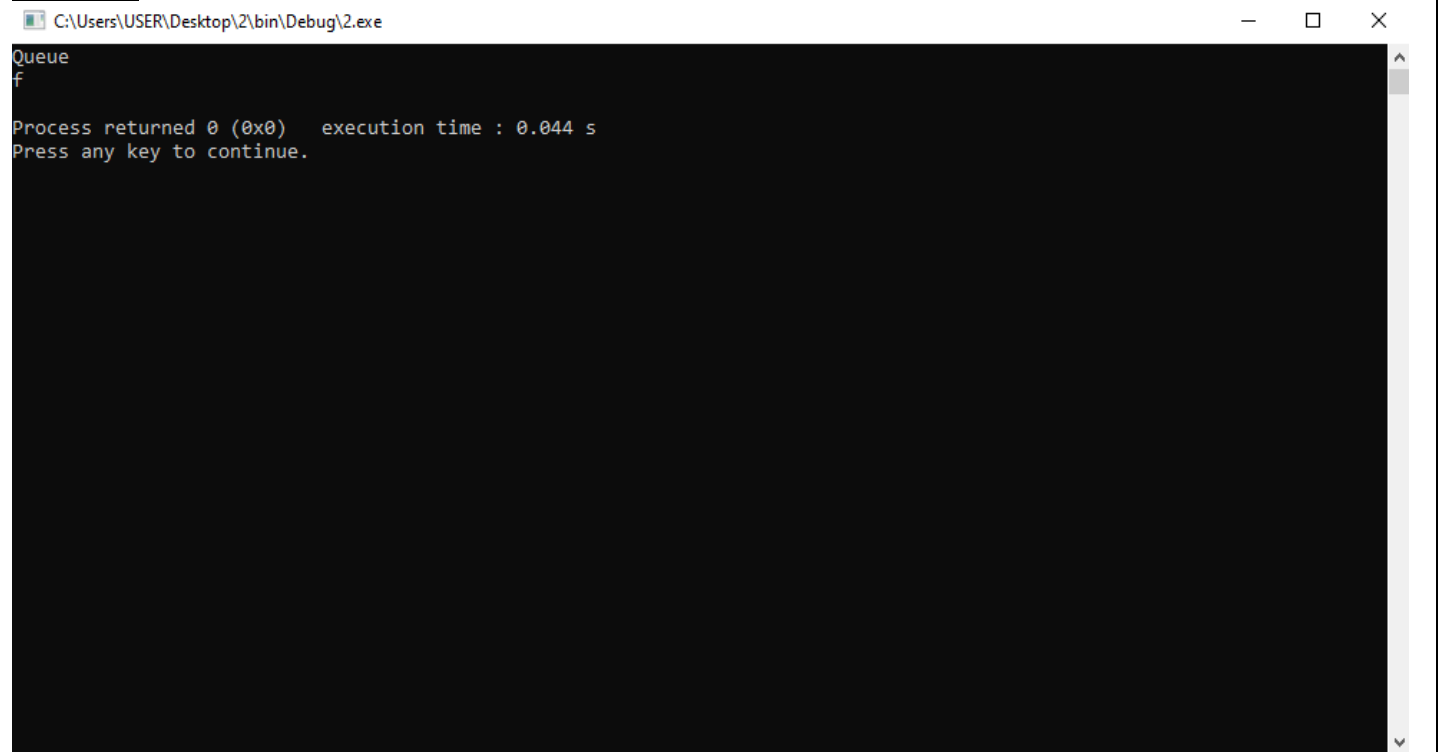
    deQueue();

    show();

    return 0;
}
```

**2. (c) Your example code's Screenshot here:-  
(insert Console Output Screenshot)**

**Answer:-**



The screenshot shows a Windows console window titled "C:\Users\USER\Desktop\2\bin\Debug\2.exe". The console output is as follows:

```
Queue  
f  
  
Process returned 0 (0x0)   execution time : 0.044 s  
Press any key to continue.
```