**TITLE OF LAB: (STACKS AND QUEUES)**

**LAB REPORT NO.08**



**Spring 2022**

**CSE-210L Data Structures and Algorithm Lab**

Submitted by

Name:: **Safi Ullah Khan**

Registration No. **20PWCSE1943**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr. Khurram Shehzad Khattak**

(Friday, July 29th, 2022)

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**OBJECTIVES OF THE LAB**

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

In this lab, we will implement and perform basic operations on Stacks and Queues data structures.

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

## **Task 01**

Implement stacks with following operations using arrays using Arrays

1. Creation
2. Pop
3. Push
4. Top
5. Size
6. Empty

**Source Code:**

**#include<iostream>**

**#include<stdlib.h>**

**#include<stdio.h> using namespace std;**

**int A[100];**

**int top=-1;**

**void Push(int z)**

**{**

**if (top==9)**

**{**

**cout<<"Error Array is Full";**

**return;**

**}**

**A[++top]=z;**

**}**

**void Pop()**

**{**

**if (top==-1)**

**{**

**cout<<"Error Array is Empty"; return;**

**}**

**cout<<A[top]<<" ";**

**top--;**

**}**

**int Top()**

**{**

**return A[top];**

**}**

**bool isfull()**

**{**

**if(top==9)**

**return true; else**

**return false;**

**}**

**bool isempty()**

**{**

**if(top==-1)**

**return true;**

**else**

**return false;**

**}**

**int main()**

**{**

**for(int i=0; i<10; i++)**

**{**

**Push(rand()%10);**

**}**

**cout<<"Array is Full "<<isfull()<<endl;**

**cout<<"\nTop Element "<<Top()<<endl;**

**for(int i=0; i<9; i++)**

**Pop();**

**}**

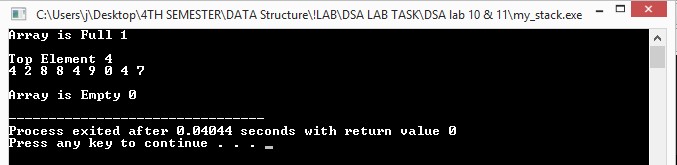
**cout<<endl;**

**cout<<"\nArray is Empty "<<isempty()<<endl;**

**return 0;**

**}**

**Screenshot Output:**



## **Task 02**

Implement stacks with following operations using linked Lists

1. Creation
2. Pop
3. Push
4. Top
5. Size
6. Empty

**Source Code:**

**#include<iostream>**

**#include<stdio.h>**

**#include<stdlib.h> using namespace std;**

**struct node**

**{**

**int data;**

**node \*link;**

**};**

**node \*top=NULL; void Push(int z)**

**{ node \*temp = (node \*)malloc(sizeof(node));**   **temp->data=z;**

**temp->link=top;**

**top=temp;**

**}**

**void Pop()**

**{**

**node \*temp;**

**if (top==NULL)**

**{**

**return;**

**}**

**temp=top;**

**cout<<top->data<<" ";**

**top=top->link;**

**delete temp;**

**} int Top()**

**{**

**return top->data;**

**}**

**bool full\_empty()**

**{**

**if(top==NULL)**

**{**

**Return true;**

**}**

**else**

**{**

**return false;**

**}**

**}**

**int main()**

**{**

**for(int i=0; i<10; i++)**

**{**

**Push(rand()%10);**

**}**

**cout<<"Top Element Data :: "<<Top();**

**cout<<endl;**

**for(int i=0; i<10; i++)**

**{**

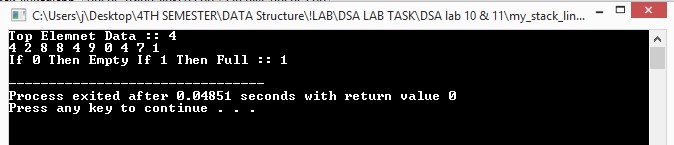
**Pop();**

**}**

**cout<<"\nIf 0 Then Empty If 1 Then Full :: "<<full\_empty()<<endl; return 0;**

**}**

**Screenshot Output:**



## 

## **Task 03**

Implement queues with following operations using arrays

1. Creation
2. Enqueue
3. Dequeue
4. Front
5. Rear
6. Size
7. Empty

**Source Code:**

**#include<iostream>**

**#include<stdlib.h>**  **using namespace std;**

**struct node**

**{**

**int data;**

**node \*next;**

**};**

**node \*front=NULL; node \*rear=NULL; void Enqueue(int x)**

**{**

**node \*temp=(node \*)malloc(sizeof(node));**

**temp->data=x;**

**temp->next=NULL;**

**if(front==NULL && rear==NULL)**

**{**

**front=rear=temp; return;**

**}**

**rear->next=temp;**

**rear=temp;**

**} void**

**Dequeue()**

**{**

**node \*temp=front;**

**if(front==NULL)**

**{**

**return;**

**}**

**if(front==rear)**

**{**

**front=rear=NULL;**

**}**

**else**

**{**

**front=front->next;**

**}**

**delete temp;**

**} void Print()**

**{**

**node\* temp = front;**

**while(temp != NULL)**

**{**

**cout<<temp->data<<",";**

**temp = temp->next;**

**}**

**cout<<endl;**

**}**

**int main()**

**{**

**Enqueue(2); Print();**

**Enqueue(4); Print();**

**Enqueue(6); Print();**

**Dequeue();**

**Print();**

**Enqueue(8);**

**Print();**

**Enqueue(10);**

**Print();**

**Dequeue();**

**Print();**

**Enqueue(12);**

**Print(); return 0;**

**}**

**Screenshot Output:**

