

Lab #10:

Name: Hassan Farooq Siddiqi

Roll No: BSSEM-S24-099

Section: BSSE-3A

Subject: Data Structure & Algorithm

**Code:**

#include<iostream>

using namespace std;

class Node{

    public:

    int data;

    Node\* next ;

    Node(int d = -1){

        data = d;

        next = nullptr;

    }

};

class Stack{

    Node\* top;

    public:

    Stack(){

        top = nullptr;

    }

    void push(int da){

        Node\* newNode= new Node(da);

        newNode->next = top;

        top = newNode;

    }

    void display(){

        Node\* temp  = top;

        while (temp != nullptr)

        {

            cout<<temp->data;

            if(temp->next != nullptr){

                cout<<" , ";

            }else {

                cout<<endl;

            }

            temp = temp->next;

        }

    }

    void pop(){

        Node\* temp = top->next;

        top = top->next;

        delete temp;

    }

    void push\_at\_end(int d){

        Node\* newNode= new Node(d);

        Node\* temp= top;

        while (temp->next != nullptr)

        {

            temp = temp->next;

        }

        temp->next = newNode;

    }

    void pop\_at\_end(){

        Node\* temp  = top;

        while (temp->next->next != nullptr)

        {

            temp = temp->next;

        }

        delete temp->next;

        temp->next = nullptr;

    }

};

class StackArray{

    int arr[100] ;

    int top ;

    public:

    StackArray(){

        top = -1;

    }

    void push(int d){

        if(top == 100){

            cout<<"Stack is Full.\n";

            return ;

        }

        top++;

        arr[top] = d;

    }

    void pop(){

        if(top == -1){

            cout<<"Stack is empty.\n";

            return ;

        }

        arr[top] = 0;

        top--;

    }

    void display(){

        if(top == -1){

            cout<<"Stack is Empty.\n";

            return ;

        }

        for (int  i = 0; i <= top; i++)

        {

            cout<<arr[i];

            if(i != top){

                cout<<" <> ";

            }

        }

        cout<<"\n";

    }

    int getStackLength(){

        return top +1 ;

    }

    int getStackSize(){

        return 100;

    }

    void push\_at\_End(int d){

        top++;

        arr[top] = d;

        int curr = 0;

        int prev = 0;

        for(int i = 0 ; i<=top ; i++){

            if(i == 0 ){

                prev = arr[i];

                arr[i] = arr[top];

            }else {

                curr = arr[i];

                arr[i] = prev;

                prev = curr;

            }

        }

    }

    void pop\_at\_End(){

        if(top != -1){

            for (int  i = 0; i <= top; i++)

            {

                arr[i] = arr[i+1];

            }

            top--;

        }

    }

};

int main(){

    cout<<"..............Stack With LinkedList.............:\n";

    Stack s;

    s.push(8);

    s.push(2);

    s.push(3);

    s.push(89);

    s.push(81);

    s.push(1);

    s.display();

    cout<<"\nAfter Pushing at End.\n";

    s.push\_at\_end(77);

    s.display();

    cout<<"\nAfter Poping at end\n";

    s.pop\_at\_end();

    s.display();

    cout<<"\nNormal Poping.\n";

    s.pop();

    s.display();

    StackArray sa;

    cout<<"\n\n............Stack With Array:............\n";

    sa.push(8);

    sa.push(7);

    sa.push(6);

    sa.push(5);

    sa.push(4);

    sa.push(3);

    sa.display();

    cout<<"\nAfter Pushing at end.\n";

    sa.push\_at\_End(9);

    sa.display();

    cout<<"\nAfter Poping at end.\n";

    sa.pop\_at\_End();

    sa.display();

    cout<<"\nNormal Poping.\n";

    sa.pop();

    sa.display();

    return 0;

}

**Output:**

