using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BT6

{

internal class Program

{

public class ArrayStack

{

private int[] stack;

private int top;

public ArrayStack(int size)

{

stack = new int[size];

top = -1;

}

public bool IsEmpty()

{

return (top == -1);

}

public bool IsFull()

{

return (top == stack.Length - 1);

}

public bool Push(int x)

{

if (IsFull())

return false;

stack[++top] = x;

return true;

}

public bool Pop(out int outItem)

{

if (IsEmpty())

{

outItem = default(int);

return false;

}

outItem = stack[top--];

return true;

}

public bool Top(out int topItem)

{

if (IsEmpty())

{

topItem = default(int);

return false;

}

topItem = stack[top];

return true;

}

}

public class Node

{

public int Data { get; set; }

public Node Next { get; set; }

}

public class ListStack

{

private Node top;

public ListStack()

{

top = null;

}

public bool IsEmpty()

{

return (top == null);

}

public void Push(int x)

{

Node newNode = new Node { Data = x };

newNode.Next = top;

top = newNode;

}

public bool Pop(out int outItem)

{

if (IsEmpty())

{

outItem = default(int);

return false;

}

outItem = top.Data;

top = top.Next;

return true;

}

public bool Top(out int topItem)

{

if (IsEmpty())

{

topItem = default(int);

return false;

}

topItem = top.Data;

return true;

}

}

static void Main(string[] args)

{

// Test ArrayStack

ArrayStack arrayStack = new ArrayStack(10);

int item;

Console.WriteLine("Testing ArrayStack:");

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

{

arrayStack.Push(i);

Console.WriteLine($"Pushed {i} into the stack");

}

arrayStack.Top(out item);

Console.WriteLine($"Top item is: {item}");

while (!arrayStack.IsEmpty())

{

arrayStack.Pop(out item);

Console.WriteLine($"Popped {item} from the stack");

}

// Test ListStack

ListStack listStack = new ListStack();

Console.WriteLine("\nTesting ListStack:");

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

{

listStack.Push(i);

Console.WriteLine($"Pushed {i} into the stack");

}

listStack.Top(out item);

Console.WriteLine($"Top item is: {item}");

while (!listStack.IsEmpty())

{

listStack.Pop(out item);

Console.WriteLine($"Popped {item} from the stack");

}

}

}

}