namespace \_09.Collection\_Hierarchy.Controllers

{

using Interfaces;

using Models;

using System;

using System.Text;

public class Engine

{

private IAddCollection<string> addcollection;

private IAddRemoveCollection<string> addRemoveCollection;

private IMyList<string> myList;

private StringBuilder resultingOutput;

public Engine()

{

this.addcollection = new AddCollection<string>();

this.addRemoveCollection = new AddRemoveCollection<string>();

this.myList = new MyList<string>();

this.resultingOutput = new StringBuilder();

}

public void Run()

{

var input = Console.ReadLine().Split();

this.FillCollection(ref input, this.addcollection);

this.FillCollection(ref input, this.addRemoveCollection);

this.FillCollection(ref input, this.myList);

var numberOfRemovals = int.Parse(Console.ReadLine());

this.RemoveOperation(numberOfRemovals, this.addRemoveCollection);

this.RemoveOperation(numberOfRemovals, this.myList);

Console.WriteLine(this.resultingOutput.ToString().Trim());

}

private void RemoveOperation<T>(int numberOfRemovals, IAddRemoveCollection<T> collection)

{

while (numberOfRemovals > 0)

{

var removedElement = collection.Remove();

this.resultingOutput.Append($"{removedElement} ");

numberOfRemovals--;

}

this.resultingOutput

.Remove(this.resultingOutput.Length - 1, 1)

.AppendLine();

}

private void FillCollection(ref string[] input, IAddCollection<string> collection)

{

foreach (var str in input)

{

var index = collection.Add(str);

this.resultingOutput.Append($"{index} ");

}

this.resultingOutput

.Remove(this.resultingOutput.Length - 1, 1)

.AppendLine();

}

}

}