|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  namespace ListOperations  {  class Program  {  static void Main(string[] args)  {  List<int> numbers = Console.  ReadLine()  .Split()  .Select(int.Parse)  .ToList();  string input;  while ((input = Console.ReadLine()) != "End")  {  string[] commands = input  .Split()  .ToArray();  switch (commands[0])  {  case "Add":  int numberToBeAdded = int.Parse(commands[1]);  AddingNumberAtTheEnd(numbers, numberToBeAdded);  break;  case "Insert":  int numberToBeInserted = int.Parse(commands[1]);  int position = int.Parse(commands[2]);  if (position < 0 || position > numbers.Count - 1)  {  Console.WriteLine("Invalid index");  continue;  }  InsertingNumberByPosition(numbers, numberToBeInserted, position);  break;  case "Remove":  int index = int.Parse(commands[1]);  if (index < 0 || index > numbers.Count - 1)  {  Console.WriteLine("Invalid index");  continue;  }  RemovingIndex(numbers, index);  break;  case "Shift":  switch (commands[1])  {  case "left":  int timesToLeft = int.Parse(commands[2]);  ShiftingToTheLeft(numbers, timesToLeft);  break;  case "right":  int timesToRight = int.Parse(commands[2]);  ShiftingToTheRight(numbers, timesToRight);  break;  }  break;  }  }  Console.WriteLine(string.Join(" ", numbers));  }  static void AddingNumberAtTheEnd(List<int> numbers, int number)  {  numbers.Add(number);  }  static void InsertingNumberByPosition(List<int> numbers, int number, int index)  {  numbers.Insert(index, number);  }  static void RemovingIndex(List<int> numbers, int index)  {  numbers.RemoveAt(index);  }  static void ShiftingToTheLeft(List<int> numbers, int timesCount)  {  for (int i = 0; i < timesCount % numbers.Count; i++)  {  numbers.Add(numbers[0]);  numbers.RemoveAt(0);  }  }  static void ShiftingToTheRight(List<int> numbers, int timesCount)  {  for (int i = 0; i < timesCount % numbers.Count; i++)  {  numbers.Insert(0, numbers[numbers.Count - 1]);  numbers.RemoveAt(numbers.Count - 1);  }  }  }  } |