ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**Факультет программной инженерии и компьютерной техники**

# Лабораторная работа №8

Восьмая пятая

Выполнил:

Жумиков Егор Олегович

Преподаватели:

Романов Алексей Андреевич

Волчек Дмитрий Геннадьевич

Санкт-Петербург, 2019

# Оглавление

# Задача «Множество»

## Условие

Реализуйте множество с операциями «добавление ключа», «удаление ключа», «проверка существования ключа».

## Формат выходного файла

Выведите последовательно результат выполнения всех операций «?». Следуйте формату выходного файла из примера.

## Решение

```
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Text;

namespace ItmoAlgos
{
    public class Program
    {
        private static string[] _input;
        private static int _currentLineIndex;

        private static string ReadLine()
        {
            return _input[_currentLineIndex++];
        }

        public static void Main(string[] args)
        {
            ISet<long> set = new SortedSet<long>();
            var output = new StringBuilder();

            _input = File.ReadAllLines("input.txt");

            long n = long.Parse(ReadLine());

            for (long i = 0; i < n; i++)
            {
                string[] strings = ReadLine().Split();
                long v = long.Parse(strings[1]);

                switch (strings[0][0])
                {
                    case 'A':
                        set.Add(v);
                        break;

                    case 'D':
                        set.Remove(v);
                        break;

                    case '?':
                        output.AppendLine(set.Contains(v) ? "Y" : "N");
                        break;
                }
            }

            using (var sw = new StreamWriter("output.txt"))
```

```
            {
                sw.Write(output.ToString());
            }
        }
    }
}
```

### Результат

Верное решение!
Результаты работы Вашего решения

| № теста | Результат | Время, с | Память | Размер входного файла | Размер выходного файла |
|---|---|---|---|---|---|
| Max | | 1.046 | 94875648 | 11189636 | 501237 |
| 1 | OK | 0.046 | 11624448 | 43 | 9 |
| 2 | OK | 0.031 | 11313152 | 8 | 3 |

## Задача «Прошитый ассоциативный массив»

### Условие

Реализуйте прошитый ассоциативный массив.

### Формат входного файла

В первой строке входного файла находится строго положительное целое число операций $N$, не превышающее $5 \cdot 10^5$. В каждой из последующих $N$ строк находится одна из следующих операций:

- `get` $x$ — если ключ $x$ есть в множестве, выведите соответствующее ему значение, если нет, то выведите `<none>`.
- `prev` $x$ — вывести значение, соответствующее ключу, находящемуся в ассоциативном массиве, который был вставлен позже всех, но до $x$, или `<none>`, если такого нет или в массиве нет $x$.
- `next` $x$ — вывести значение, соответствующее ключу, находящемуся в ассоциативном массиве, который был вставлен раньше всех, но после $x$, или `<none>`, если такого нет или в массиве нет $x$.
- `put` $x$ $y$ — поставить в соответствие ключу $x$ значение $y$. При этом следует учесть, что:
  - если, независимо от предыстории, этого ключа на момент вставки в массиве не было, то он считается только что вставленным и оказывается самым последним среди добавленных элементов — то есть, вызов `next` с этим же ключом сразу после выполнения текущей операции `put` должен вернуть `<none>`;
  - если этот ключ уже есть в массиве, то значение необходимо изменить, и в этом случае ключ не считается вставленным еще раз, то есть, не меняет своего положения в порядке добавленных элементов.
- `delete` $x$ — удалить ключ $x$. Если ключа $x$ в ассоциативном массиве нет, то ничего делать не надо.

Ключи и значения — строки из латинских букв длиной не менее одного и не более 20 символов.

### Формат выходного файла

Выведите последовательно результат выполнения всех операций get, prev, next. Следуйте формату выходного файла из примера.

### Решение

```
using System.Collections.Generic;
```

```
using System.IO;
using System.Text;

namespace ItmoAlgos
{
    public class Program
    {
        private static string[] _input;
        private static int _currentLineIndex;

        private static string ReadLine()
        {
            return _input[_currentLineIndex++];
        }

        public static void Main(string[] args)
        {
            var dictionary = new Dictionary<string,
LinkedListNode<string>>();
            var linkedList = new LinkedList<string>();
            var output = new StringBuilder();
            var emptyNode = new LinkedListNode<string>("<none>");
            LinkedListNode<string> node;

            _input = File.ReadAllLines("input.txt");

            long n = long.Parse(ReadLine());


            for (long i = 0; i < n; i++)
            {
                string[] strings = ReadLine().Split();
                string key = strings[1];
                string value = strings.Length == 3 ? strings[2] :
null;

                switch (strings[0])
                {
                    case "put":
                        if (dictionary.TryGetValue(key, out node))
                        {
                            node.Value = value;
                        }
                        else
                        {
                            dictionary[key] =
linkedList.AddLast(value);
                        }
                        break;

                    case "get":
```

```
output.AppendLine((dictionary.GetValueOrDefault(key) ??
emptyNode).Value);
                        break;

                    case "prev":
                        output.AppendLine((dictionary[key].Previous ??
emptyNode).Value);
                        break;

                    case "next":
                        output.AppendLine((dictionary[key].Next ??
emptyNode).Value);
                        break;

                    case "delete":
                        if (dictionary.Remove(key, out node))
                        {
                            linkedList.Remove(node);
                        }
                        break;
                }
            }

            File.WriteAllText("output.txt", output.ToString());
        }
    }
}
```

Результат

Верное решение!
Результаты работы Вашего решения

| № теста | Результат | Время, с | Память | Размер входного файла | Размер выходного файла |
|---|---|---|---|---|---|
| Max | | 1.531 | 231374848 | 23499808 | 10303658 |
| 1 | OK | 0.015 | 10768384 | 158 | 26 |
| 2 | OK | 0.031 | 10698752 | 12 | 8 |
| 3 | OK | 0.031 | 10690560 | 25 | 5 |
| 4 | OK | 0.046 | 10678272 | 25 | 8 |