Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Кафедра информатики и прикладной математики

Лабораторная работа №4
Дисциплина «Алгоритмы и структуры данных»

Выполнил:
Молодецкий Арсений Алексеевич
Группа P3217

Санкт-Петербург
2018

Задание №1:

Код:

```
#include <iostream>
#include <string>

using namespace std;

#include "edx-io.hpp"
#define cin io
#define cout io


int main() {
    char operation;
    int* stack = new int[1000000];
    int pointer = -1;

    int n;
    cin >> n;

    for (int i = 0; i < n; i++) {
        cin >> operation;
        if (operation == '-') {
            cout << stack[pointer] << '\n';
            pointer--;
        }
        else {
            pointer++;
            cin >> stack[pointer];
        }
    }

    return 0;
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.156	20787200	13389454	5693807
1	OK	0.015	3448832	33	10
2	OK	0.000	3457024	11	3
3	OK	0.000	3457024	19	6
4	OK	0.000	3444736	19	6
5	OK	0.015	3440640	19	6
6	OK	0.015	3436544	96	45
7	OK	0.000	3440640	85	56
8	OK	0.000	3428352	129	11
9	OK	0.000	3436544	131	12
10	OK	0.015	3444736	859	540
11	OK	0.015	3457024	828	573
12	OK	0.015	3428352	1340	11
13	OK	0.015	3452928	1325	12
14	OK	0.000	3428352	8292	5590
15	OK	0.000	3452928	8212	5706
16	OK	0.062	3436544	13298	111
17	OK	0.000	3424256	13354	12
18	OK	0.015	3461120	82372	56548
19	OK	0.000	3469312	82000	56993
20	OK	0.000	3563520	132796	1134
21	OK	0.000	3534848	133914	11
22	OK	0.031	4206592	819651	569557
23	OK	0.031	4415488	819689	569681
24	OK	0.000	5120000	1328670	11294
25	OK	0.015	5111808	1338543	11
26	OK	0.156	11591680	8196274	5693035
27	OK	0.156	13570048	8193816	5693807
28	OK	0.078	20602880	13286863	112020
29	OK	0.046	20758528	13389454	11
30	OK	0.046	20787200	13388564	11

Задание №2:

Код:

```
#include <iostream>
#include <string>

using namespace std;
```

```
#include "edx-io.hpp"
#define cin io
#define cout io

int main() {
    char operation;
    int* queue = new int[1000000];
    int pointerEnd = -1;
    int pointerStart = 0;
    int n;
    cin >> n;

    for (int i = 0; i < n; i++) {
        cin >> operation;
        if (operation == '-') {
            cout << queue[pointerStart] << '\n';
            pointerStart++;
        }
        else {
            pointerEnd++;
            cin >> queue[pointerEnd];
        }
    }

    return 0;
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.171	20754432	13389454	5693807
1	OK	0.000	3424256	20	7
2	OK	0.015	3432448	11	3
3	OK	0.000	3448832	19	6
4	OK	0.015	3448832	19	6
5	OK	0.000	3457024	96	45
6	OK	0.000	3432448	85	56
7	OK	0.000	3440640	129	12
8	OK	0.000	3444736	131	12
9	OK	0.000	3432448	859	538
10	OK	0.000	3424256	828	573
11	OK	0.000	3428352	1340	12
12	OK	0.046	3428352	1325	12
13	OK	0.000	3452928	8292	5589
14	OK	0.000	3444736	8212	5706
15	OK	0.000	3428352	13298	115
16	OK	0.015	3420160	13354	12
17	OK	0.015	3493888	82372	56552
18	OK	0.015	3457024	82000	56993
19	OK	0.000	3543040	132796	1124
20	OK	0.000	3530752	133914	12
21	OK	0.015	4407296	819651	569553
22	OK	0.015	4386816	819689	569681
23	OK	0.015	5103616	1328670	11296
24	OK	0.015	5128192	1338543	12
25	OK	0.171	13586432	8196274	5693025
26	OK	0.171	13561856	8193816	5693807
27	OK	0.046	20619264	13286863	112110
28	OK	0.062	20746240	13389454	10
29	OK	0.062	20754432	13388564	11

Задание №3

Код

```
#include <iostream>
#include <string>

using namespace std;

#define LRB '('
```

```
#define RRB ')'
#define LSB '['
#define RSB ']'

int main() {
    char * stack = new char[1000000];
    int pointer = -1;
    int n;
    cin >> n;

    string sequence;
    for (int i = 0; i < n; i++) {
        cin >> sequence;
        bool isWrong = false;
        pointer = -1;
        for (int j = 0; j < sequence.length() && !isWrong; ++j) {
            if (sequence[j] == LSB || sequence[j] == LRB){
                pointer++;
                stack[pointer] = sequence[j];
            } else {
                if (pointer == -1){
                    isWrong = true;
                }
                else {
                    if (sequence[j] == RRB){
                        stack[pointer] == LRB ? (pointer--) : isWrong = true;
                    } else{
                        stack[pointer] == LSB ? (pointer--) : isWrong = true;
                    }
                }

            }
        }
        isWrong = isWrong || pointer > -1;
        cout << (isWrong ? "NO\n" : "YES\n");
    }

    return 0;
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.046	8441856	5000885	2133
1	OK	0.000	3477504	31	22
2	OK	0.015	3452928	15	16
3	OK	0.015	3448832	68	66
4	OK	0.015	3465216	324	256
5	OK	0.015	3452928	1541	1032
6	OK	0.015	3452928	5880	2128
7	OK	0.000	3481600	50867	2129
8	OK	0.000	3919872	500879	2110
9	OK	0.046	8417280	5000884	2120
10	OK	0.046	8441856	5000885	2133

Задание №4

Код

```
#include <iostream>
#include <string>
#include <deque>

using namespace std;


int main() {
    char operation;
    long* queue = new long[1000000];
    int pointerEnd = -1;
    int pointerStart = 0;
    int n;
    cin >> n;

    deque<long> deque;

    for (int i = 0; i < n; i++) {
        cin >> operation;
        if (operation == '-') {
            if (deque.front() == queue[pointerStart]){
                deque.pop_front();
            }
            pointerStart++;
        } else if (operation == '?'){
            cout << to_string(deque.front()) << '\n';
        } else {
            pointerEnd++;
            cin >> queue[pointerEnd];
            while (!deque.empty() && deque.back() > queue[pointerEnd]) {
                deque.pop_back();
            }
            deque.push_back(queue[pointerEnd]);
        }
    }

    return 0;
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.109	29454336	13389342	4002151
1	OK	0.000	2248704	29	10
2	OK	0.015	2248704	11	3
3	OK	0.015	2236416	22	6
4	OK	0.000	2236416	22	6
5	OK	0.015	2236416	36	9
6	OK	0.000	2236416	48	12
7	OK	0.000	2236416	76	35
8	OK	0.000	2248704	129	12
9	OK	0.000	2248704	67	48
10	OK	0.015	2248704	44	9
11	OK	0.000	2236416	45	9
12	OK	0.000	2236416	44	9
13	OK	0.000	2236416	45	9
14	OK	0.000	2248704	721	384
15	OK	0.015	2236416	1340	12
16	OK	0.000	2236416	640	407
17	OK	0.000	2236416	445	90
18	OK	0.000	2232320	456	100
19	OK	0.015	2248704	445	90
20	OK	0.000	2248704	456	100
21	OK	0.015	2248704	6616	3812
22	OK	0.000	2252800	13389	12
23	OK	0.000	2236416	6461	4008
24	OK	0.015	2248704	4896	1140
25	OK	0.000	2236416	5007	1250
26	OK	0.000	2248704	4896	1140
27	OK	0.000	2236416	5007	1250
28	OK	0.000	2248704	64907	39589
29	OK	0.015	2273280	133814	12
30	OK	0.000	2244608	64675	39996
31	OK	0.000	2265088	53897	13890
32	OK	0.000	2252800	55008	15000
33	OK	0.000	2285568	53897	13890
34	OK	0.015	2252800	55008	15000
35	OK	0.015	2600960	645271	404305
36	OK	0.000	3559424	1338956	12
37	OK	0.000	2605056	646300	400008
38	OK	0.015	3178496	588898	163890
39	OK	0.015	2625536	600009	175000
40	OK	0.015	3235840	588898	163890
41	OK	0.015	2625536	600009	175000
42	OK	0.109	9625600	6465010	4002151
43	OK	0.078	19210240	13389342	12
44	OK	0.109	9621504	6462989	4000004
45	OK	0.109	15396864	6388899	1888890
46	OK	0.093	10326016	6500010	2000000
47	OK	0.093	15360000	6388899	1888890
48	OK	0.078	10321920	6500010	2000000
49	OK	0.078	19210240	13388086	12
50	OK	0.015	2236416	55	16
51	OK	0.015	2248704	705	225
52	OK	0.000	2248704	6506	2000
53	OK	0.015	2310144	65007	20000
54	OK	0.015	3026944	650008	200000
55	OK	0.093	13262848	6675213	2000000
56	OK	0.000	2232320	117	12
57	OK	0.015	2232320	1327	12
58	OK	0.031	2248704	13417	12
59	OK	0.000	2314240	133845	12
60	OK	0.015	4685824	1339319	12
61	OK	0.093	29454336	13388955	12

Задание №5

Код

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

namespace tmp
{
    public class Program
    {
        private static StreamWriter _out;

        private static void Main(string[] args)
        {

            _out = new StreamWriter("output.txt");
            Console.SetOut(_out);
            new Quack().Run(File.ReadAllLines("input.txt"));
            DisposeIO();
        }

        private static void DisposeIO()
        {
            _out?.Dispose();
        }
    }

    public class Quack
    {
        private readonly Queue<ushort> _queue = new Queue<ushort>();

        private Dictionary<int, ushort> _registers;

        private int _cursor;

        private Dictionary<char, Action<string>> _instructions;
        private Dictionary<string, int> _labels;
        private bool _isRunning;

        public void Run(string[] lines)
        {
            _queue.Clear();

            _isRunning = true;
            DefineRegisters();
            DefineLabels(lines);
            DefineInstructions();

            while (_cursor < lines.Length && _isRunning) {
                Interpret(lines[_cursor]);
                _cursor++;
            }
        }

        private void Interpret(string s)
```

```
{
   if (char.IsDigit(s[0])) {
      Put(ushort.Parse(s));
      return;
   }

   _instructions[s[0]](s);
}

private void Print(int number)
{
   Console.WriteLine(number);
}
private void Print(char c)
{
   Console.Write(c);
}
private void changeCursor(string line, int indent)
{
   string label = new string(line.Skip(indent).ToArray());
   _cursor = _labels[label];
}
private ushort Get()
{
   return _queue.Dequeue();
}

private void Put(int value)
{
   _queue.Enqueue((ushort) (value % Int16.MaxValue));
}
private void DefineLabels(string[] lines)
{
   _labels = lines
      .Select((s, i) => (label: s, line: i))
      .Where(s => s.label[0] == ':')
      .ToDictionary(tuple => new string(tuple.label.Skip(1).ToArray()), tuple => tuple.line);
}
private void DefineRegisters()
{
   _registers = Enumerable.Range('a', 26)
      .ToDictionary(x => x, y => (ushort) 0);
}
private void DefineInstructions()
{
   _instructions = new Dictionary<char, Action<string>>
   {
      ['+'] = s => { Put(Get() + Get()); },
      ['-'] = s => { Put(Get() - Get()); },
      ['*'] = s => { Put(Get() * Get()); },
      ['/'] = s => {
         var a = Get();
         var b = Get();
         Put(b == 0 ? 0 : a / b);
      },
      ['%'] = s => {
         var a = Get();
         var b = Get();
         Put(b == 0 ? 0 : a % b);
      },
```

```
            ['>'] = s => { _registers[s[1]] = Get(); },
            ['<'] = s => { Put(_registers[s[1]]); },
            ['P'] = s => { Print(s.Length == 1 ? Get() : _registers[s[1]]); },
            ['C'] = s => { Print((char) ((s.Length == 1 ? Get() : _registers[s[1]]) % 256)); },
            [':'] = s => { },
            ['J'] = s => { changeCursor(s,1); },
            ['Z'] = s => {
               if (_registers[s[1]] == 0) {
                  changeCursor(s,2);
               }
            },
            ['E'] = s => {
               if (_registers[s[1]] == _registers[s[2]]) {
                  changeCursor(s,3);
               }
            },
            ['G'] = s => {
               if (_registers[s[1]] > _registers[s[2]]) {
                  changeCursor(s,3);
               }
            },
            ['Q'] = s => { _isRunning = false; }
         };


      }
   }
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.156	31842304	1349803	250850
1	OK	0.062	13164544	69	6
2	OK	0.062	13209600	232	218
3	OK	0.062	12902400	3	0
4	OK	0.062	12902400	100	19
5	OK	0.062	14336000	56	58890
6	OK	0.046	14307328	67	30000
7	OK	0.093	14299136	67	30000
8	OK	0.062	14340096	55	30000
9	OK	0.046	12902400	461	60
10	OK	0.046	13111296	11235	21
11	OK	0.062	13398016	23748	42
12	OK	0.046	14086144	66906	8905
13	OK	0.062	13004800	7332	954
14	OK	0.046	13004800	4611	602
15	OK	0.062	13594624	37968	5424
16	OK	0.046	12972032	14	2
17	OK	0.046	12910592	70	10
18	OK	0.046	12918784	350	50
19	OK	0.062	12967936	1750	250
20	OK	0.046	13070336	8750	1250
21	OK	0.062	13672448	43750	6250
22	OK	0.062	16838656	218750	31250
23	OK	0.046	13553664	34606	4721
24	OK	0.078	20467712	683180	7
25	OK	0.078	20443136	683102	0
26	OK	0.156	31842304	1349803	0
27	OK	0.078	21053440	491572	247791
28	OK	0.093	21094400	491488	249618
29	OK	0.093	21114880	491600	249600
30	OK	0.078	21094400	491502	250850
31	OK	0.093	21061632	491416	249477
32	OK	0.093	21082112	491520	250262
33	OK	0.093	21106688	491317	246859
34	OK	0.078	21073920	491514	248199
35	OK	0.078	21110784	491557	249601