

Университет ИТМО  
Факультет программной инженерии и компьютерной техники  
Алгоритмы и структуры данных

## Лабораторная работа №4

Выполнил:  
Ярошевич Александр Р3217

Санкт-Петербург  
2019

## Задание №1

### Стек

1.0 из 1.0 балла (оценивается)

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте работу стека. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо "+  $N$ ", либо "-". Команда "+  $N$ " означает добавление в стек числа  $N$ , по модулю не превышающего  $10^9$ . Команда "-" означает изъятие элемента из стека. Гарантируется, что не происходит извлечения из пустого стека. Гарантируется, что размер стека в процессе выполнения команд не превысит  $10^6$  элементов.

#### Формат входного файла

В первой строке входного файла содержится  $M$  ( $1 \leq M \leq 10^6$ ) — число команд. Каждая последующая строка исходного файла содержит ровно одну команду.

#### Формат выходного файла

Выведите числа, которые удаляются из стека с помощью команды "-", по одному в каждой строке. Числа нужно выводить в том порядке, в котором они были извлечены из стека. Гарантируется, что изъятий из пустого стека не производится.

#### Пример

input.txt	output.txt
6	10
+ 1	1234
+ 10	
-	
+ 2	
+ 1234	
-	

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.140	19202048	13389454	5693807
1	OK	0.000	2240512	33	10
2	OK	0.000	2228224	11	3
3	OK	0.015	2228224	19	6
4	OK	0.031	2224128	19	6
5	OK	0.015	2228224	19	6
6	OK	0.000	2228224	96	45
7	OK	0.000	2228224	85	56
8	OK	0.000	2228224	129	11
9	OK	0.000	2224128	131	12
10	OK	0.000	2228224	859	540
11	OK	0.000	2232320	828	573
12	OK	0.000	2228224	1340	11
13	OK	0.015	2228224	1325	12
14	OK	0.000	2244608	8292	5590
15	OK	0.000	2228224	8212	5706
16	OK	0.000	2228224	13298	111
17	OK	0.000	2228224	13354	12
18	OK	0.000	2228224	82372	56548
19	OK	0.015	2244608	82000	56993
20	OK	0.015	2265088	132796	1134
21	OK	0.000	2265088	133914	11
22	OK	0.015	2637824	819651	569557
23	OK	0.015	2838528	819689	569681
24	OK	0.015	3538944	1328670	11294
25	OK	0.015	3551232	1338543	11
26	OK	0.140	10018816	8196274	5693035
27	OK	0.140	12013568	8193816	5693807
28	OK	0.062	19025920	13286863	112020
29	OK	0.062	19202048	13389454	11
30	OK	0.062	19202048	13388564	11

## Код

```
#include
"edx-io.hpp"
```

```
int week4_1() {
    char operation;
```

```
auto stack = new int[1000000];

int n;

io >> n;


for (int i = 0; i < n; i++) {
    io >> operation;

    if (operation == '-') {
        io << *--stack <<
'\n';

    }

    else {
        io >> *(stack++);
    }
}


return 0;
}
```

## **Задача 2**

## Очередь

1.0 из 1.0 балла (оценивается)

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте работу очереди. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо «+  $N$ », либо «-». Команда «+  $N$ » означает добавление в очередь числа  $N$ , по модулю не превышающего  $10^9$ . Команда «-» означает изъятие элемента из очереди. Гарантируется, что размер очереди в процессе выполнения команд не превысит  $10^6$  элементов.

### Формат входного файла

В первой строке содержится  $M$  ( $1 \leq M \leq 10^6$ ) — число команд. В последующих строках содержатся команды, по одной в каждой строке.

### Формат выходного файла

Выведите числа, которые удаляются из очереди с помощью команды «-», по одному в каждой строке. Числа нужно выводить в том порядке, в котором они были извлечены из очереди. Гарантируется, что извлечения из пустой очереди не производится.

### Пример

input.txt	output.txt
4	1
+ 1	10
+ 10	
-	
-	

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.156	19202048	13389454	5693807
1	OK	0.000	2228224	20	7
2	OK	0.000	2228224	11	3
3	OK	0.015	2228224	19	6
4	OK	0.015	2228224	19	6
5	OK	0.000	2228224	96	45
6	OK	0.000	2224128	85	56
7	OK	0.000	2220032	129	12
8	OK	0.000	2240512	131	12
9	OK	0.000	2244608	859	538
10	OK	0.015	2240512	828	573
11	OK	0.015	2240512	1340	12
12	OK	0.000	2228224	1325	12
13	OK	0.000	2240512	8292	5589
14	OK	0.015	2224128	8212	5706
15	OK	0.000	2240512	13298	115
16	OK	0.015	2236416	13354	12
17	OK	0.000	2244608	82372	56552
18	OK	0.015	2244608	82000	56993
19	OK	0.000	2260992	132796	1124
20	OK	0.015	2260992	133914	12
21	OK	0.015	2838528	819651	569553
22	OK	0.015	2838528	819689	569681
23	OK	0.015	3538944	1328670	11296
24	OK	0.015	3551232	1338543	12
25	OK	0.156	12017664	8196274	5693025
26	OK	0.140	12013568	8193816	5693807
27	OK	0.062	19062784	13286863	112110
28	OK	0.062	19202048	13389454	10
29	OK	0.062	19202048	13388564	11

```
#include
"edx-io.hpp"
```

```
int main(){
    char operation;
    auto queue = new int[1000000];
    int endPtr = 0;
```

```
int startPtr = 0;

int n;

io >> n;

for (int i = 0; i < n; i++) {
    io >> operation;

    if (operation == '-') {
        io << queue[startPtr] <<
'\n';
        startPtr++;
    }
    else {
        io >> (queue[endPtr]);
        endPtr++;
    }
}

return 0;
}
```

### **Задача 3**



## Скобочная последовательность

1.0 из 1.0 балла (оценивается)

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Последовательность  $A$ , состоящую из символов из множества «(», «)», «[» и «]», назовем *правильной скобочной последовательностью*, если выполняется одно из следующих утверждений:

- $A$  — пустая последовательность;
- первый символ последовательности  $A$  — это «(», и в этой последовательности существует такой символ «)», что последовательность можно представить как  $A = (B)C$ , где  $B$  и  $C$  — правильные скобочные последовательности;
- первый символ последовательности  $A$  — это «[», и в этой последовательности существует такой символ «]», что последовательность можно представить как  $A = [B]C$ , где  $B$  и  $C$  — правильные скобочные последовательности.

Так, например, последовательности «(())» и «()[]» являются правильными скобочными последовательностями, а последовательности «[]» и «((» таковыми не являются.

Входной файл содержит несколько строк, каждая из которых содержит последовательность символов «(», «)», «[» и «]». Для каждой из этих строк выясните, является ли она правильной скобочной последовательностью.

### Формат входного файла

Первая строка входного файла содержит число  $N$  ( $1 \leq N \leq 500$ ) - число скобочных последовательностей, которые необходимо проверить. Каждая из следующих  $N$  строк содержит скобочную последовательность длиной от 1 до  $10^4$  включительно. В каждой из последовательностей присутствуют только скобки указанных выше видов.

### Формат выходного файла

Для каждой строки входного файла выведите в выходной файл «YES», если соответствующая последовательность является правильной скобочной последовательностью, или «NO», если не является.

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.031	6844416	5000885	2133
1	OK	0.015	2240512	31	22
2	OK	0.000	2240512	15	16
3	OK	0.000	2240512	68	66
4	OK	0.000	2236416	324	256
5	OK	0.015	2248704	1541	1032
6	OK	0.000	2236416	5880	2128
7	OK	0.000	2248704	50867	2129
8	OK	0.000	2330624	500879	2110
9	OK	0.015	6844416	5000884	2120
10	OK	0.031	6844416	5000885	2133



```

#include
<string>

#include "edx-io.hpp"

using namespace std;

int main() {

    auto stack = new char[1000000];

    int ptr;

    int n;

    io >> n;

    string str;

    for (int i = 0; i < n; i++) {

        ptr = -1;

        io >> str;

        bool isWrong = false;

        int len = str.length();

        if (len % 2 == 1)

            isWrong = true;

        for (int j = 0; j < len && !isWrong; ++j) {

            if (str[j] == '(' || str[j] == '['){

                ptr++;

                stack[ptr] = str[j];

            } else {

                if (ptr == -1){

                    isWrong = true;

                }

                else {

                    if (str[j] == '){

                        stack[ptr] == '(' ? (ptr--): isWrong =

true;

```

```

        } else{
            stack[ptr] == '[' ? (ptr--) : isWrong =
true;
        }
    }

}

}

}

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

return 0;
}

```

#### **Задача 4**

## Очередь с минимумом

2.0 из 2.0 баллов (оценивается)

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте работу очереди. В дополнение к стандартным операциям очереди, необходимо также отвечать на запрос о минимальном элементе из тех, которые сейчас находятся в очереди. Для каждой операции запроса минимального элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо «+  $N$ », либо «-», либо «?». Команда «+  $N$ » означает добавление в очередь числа  $N$ , по модулю не превышающего  $10^9$ . Команда «-» означает изъятие элемента из очереди. Команда «?» означает запрос на поиск минимального элемента в очереди.

### Формат входного файла

В первой строке содержится  $M$  ( $1 \leq M \leq 10^6$ ) — число команд. В последующих строках содержатся команды, по одной в каждой строке.

### Формат выходного файла

Для каждой операции поиска минимума в очереди выведите её результат. Результаты должны быть выведены в том порядке, в котором эти операции встречаются во входном файле. Гарантируется, что операций извлечения или поиска минимума для пустой очереди не производится.