

**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 №4

Неделя четвертая

Выполнил:

Жумиков Егор Олегович

Преподаватели:

Романов Алексей Андреевич

Волчек Дмитрий Геннадьевич

Оглавление

Задача «Сортировка целых чисел »	3
Условие.....	3
Формат входного файла.....	3
Формат выходного файла	3
Решение	3
Результат	4
Задача «Цифровая сортировка».....	5
Условие.....	5
Формат входного файла.....	5
Формат выходного файла	5
Решение	5
Результат	6

Задача «Стек»

Условие

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.

Решение

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

using namespace std;

#ifdef LOCAL

#define cin std::cin
#define cout std::cout

#else

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

#endif

int main() {
    int n;
    long t;
    char cmd;
    long *stack = new long[1e6];
    long ptr = 0;

    cin >> n;

    for (int i = 0; i < n; i++) {
        cin >> cmd;
        if (cmd == '+') {
```

```

        cin >> stack[ptr++];
    }
    else {
        cout << stack[--ptr] << '\n';
    }
}

return 0;
}

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.156	19202048	13389454	5693807
1	OK	0.015	2228224	33	10
2	OK	0.015	2228224	11	3
3	OK	0.015	2240512	19	6
4	OK	0.000	2224128	19	6
5	OK	0.000	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	2228224	131	12
10	OK	0.000	2224128	859	540
11	OK	0.000	2228224	828	573
12	OK	0.000	2224128	1340	11
13	OK	0.000	2228224	1325	12
14	OK	0.000	2224128	8292	5590
15	OK	0.015	2228224	8212	5706
16	OK	0.015	2228224	13298	111
17	OK	0.000	2240512	13354	12
18	OK	0.000	2228224	82372	56548
19	OK	0.000	2240512	82000	56993
20	OK	0.000	2277376	132796	1134

21	OK	0.015	2265088	133914	11
22	OK	0.015	2641920	819651	569557
23	OK	0.000	2834432	819689	569681
24	OK	0.000	3538944	1328670	11294
25	OK	0.000	3551232	1338543	11
26	OK	0.156	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

Задача «Очередь»

Условие

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.

Решение

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

using namespace std;

#ifdef LOCAL

#define cin std::cin
#define cout std::cout

#else

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

```
#endif
```

```
int main() {  
    int n;  
    long t;  
    char cmd;  
    long *queue = new long[1 + 1e6];  
    long fptr = 0, lptr = 0;  
  
    cin >> n;  
  
    for (int i = 0; i < n; i++) {  
        cin >> cmd;  
        if (cmd == '+') {  
            cin >> queue[fptr++];  
        }  
        else {  
            cout << queue[lptr++] << '\n';  
        }  
    }  
  
    return 0;  
}
```

Результат

№ 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	0.000	2224128	19	6
4	OK	0.000	2228224	19	6
5	OK	0.015	2228224	96	45
6	OK	0.000	2228224	85	56
7	OK	0.015	2224128	129	12
8	OK	0.000	2224128	131	12
9	OK	0.000	2228224	859	538
10	OK	0.000	2228224	828	573
11	OK	0.000	2224128	1340	12
12	OK	0.000	2240512	1325	12
13	OK	0.000	2224128	8292	5589
14	OK	0.000	2224128	8212	5706

15	OK	0.000	2228224	13298	115
16	OK	0.000	2240512	13354	12
17	OK	0.000	2244608	82372	56552
18	OK	0.000	2244608	82000	56993
19	OK	0.015	2273280	132796	1124
20	OK	0.000	2265088	133914	12
21	OK	0.015	2838528	819651	569553
22	OK	0.015	2834432	819689	569681
23	OK	0.015	3543040	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	0.046	19202048	13389454	10
29	OK	0.078	19202048	13388564	11

Задача «Скобочная последовательность»

Условие

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.

Решение

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

using namespace std;

#ifdef LOCAL

#define cin std::cin
#define cout std::cout

#else

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

#endif

int main() {
    int n;
    long t;
    string cmd;
    char *queue = new char[1e4];

    cin >> n;
    long ptr;

    for (int i = 0; i < n; i++) {
        long ptr = 0;
        cin >> cmd;

        for (int j = 0; j < cmd.size(); j++) {
            switch (cmd[j])
            {
                case '(': case '[':
                    queue[ptr++] = cmd[j];
                    break;

                case ')': case ']':
```



```

        if (ptr ≤ 0 || queue[--ptr] ≠ (cmd[j] = '))' ?
'(' : '[')) {
        //cout << ptr << " " << queue[ptr] << " " <<
cmd[j] << " ";
        cout << "NO\n";
        goto end;
    }
    break;
}
}

cout << (ptr ? "NO\n" : "YES\n");

end;;
}

return 0;
}

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.031	6832128	5000885	2133
1	OK	0.000	2236416	31	22
2	OK	0.000	2248704	15	16
3	OK	0.000	2232320	68	66
4	OK	0.000	2248704	324	256
5	OK	0.000	2236416	1541	1032
6	OK	0.000	2232320	5880	2128
7	OK	0.015	2252800	50867	2129
8	OK	0.015	2326528	500879	2110
9	OK	0.031	6832128	5000884	2120
10	OK	0.031	6832128	5000885	2133

Задача «Очередь с минимумом»

Условие

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 109. 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.

Решение

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

using namespace std;

#ifdef LOCAL

#define cin std::cin
#define cout std::cout

#else

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

#endif

int main() {
    int n;
    long t;
    char cmd;
    queue<long> q;
    deque<long> d;

    cin >> n;

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

        switch (cmd)
        {
            case '+':
                cin >> t;
                q.push(t);
                while (!d.empty() && d.back() > t) {
                    d.pop_back();
                }
                d.push_back(t);
                break;

            case '-':
```

```

        if (d.front() == q.front()) { d.pop_front(); }
        q.pop();
        break;

    case '?':
        cout << d.front() << '\n';
        break;
    }
}

return 0;
}

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.156	35635200	13389342	4002151
1	OK	0.015	2224128	29	10
2	OK	0.000	2236416	11	3
3	OK	0.000	2220032	22	6
4	OK	0.000	2224128	22	6
5	OK	0.000	2224128	36	9
6	OK	0.000	2224128	48	12
7	OK	0.000	2224128	76	35
8	OK	0.000	2220032	129	12
9	OK	0.000	2236416	67	48
10	OK	0.000	2224128	44	9
11	OK	0.000	2220032	45	9
12	OK	0.015	2220032	44	9
13	OK	0.000	2224128	45	9
14	OK	0.000	2224128	721	384
15	OK	0.015	2228224	1340	12
16	OK	0.015	2224128	640	407
17	OK	0.000	2232320	445	90
18	OK	0.000	2220032	456	100
19	OK	0.015	2224128	445	90
20	OK	0.000	2224128	456	100
21	OK	0.015	2236416	6616	3812
22	OK	0.015	2240512	13389	12
23	OK	0.000	2236416	6461	4008
24	OK	0.000	2240512	4896	1140

25	OK	0.000	2236416	5007	1250
26	OK	0.000	2240512	4896	1140
27	OK	0.000	2248704	5007	1250
28	OK	0.000	2228224	64907	39589
29	OK	0.015	2248704	133814	12
30	OK	0.000	2265088	64675	39996
31	OK	0.000	2260992	53897	13890
32	OK	0.000	2240512	55008	15000
33	OK	0.000	2248704	53897	13890
34	OK	0.015	2236416	55008	15000
35	OK	0.015	2473984	645271	404305
36	OK	0.015	4280320	1338956	12
37	OK	0.015	2945024	646300	400008
38	OK	0.015	3645440	588898	163890
39	OK	0.015	3047424	600009	175000
40	OK	0.015	3645440	588898	163890
41	OK	0.015	2985984	600009	175000
42	OK	0.125	8318976	6465010	4002151
43	OK	0.140	25485312	13389342	12
44	OK	0.125	12132352	6462989	4000004
45	OK	0.125	18456576	6388899	1888890
46	OK	0.109	13500416	6500010	2000000
47	OK	0.140	18456576	6388899	1888890
48	OK	0.093	13463552	6500010	2000000
49	OK	0.125	25444352	13388086	12
50	OK	0.000	2236416	55	16
51	OK	0.015	2224128	705	225
52	OK	0.031	2236416	6506	2000
53	OK	0.015	2265088	65007	20000
54	OK	0.015	3170304	650008	200000
55	OK	0.109	13914112	6675213	2000000
56	OK	0.015	2236416	117	12
57	OK	0.000	2252800	1327	12
58	OK	0.015	2252800	13417	12
59	OK	0.000	2297856	133845	12

60	OK	0.015	5439488	1339319	12
61	OK	0.156	35635200	13388955	12

Задача «Quack»

Условие

Язык Quack — забавный язык, который фигурирует в одной из задач с Internet Problem Solving Contest. В этой задаче вам требуется написать интерпретатор языка Quack.

Виртуальная машина, на которой исполняется программа на языке Quack, имеет внутри себя очередь, содержащую целые числа по модулю 65536 (то есть, числа от 0 до 65535, соответствующие беззнаковому 16-битному целому типу). Слово `get` в описании операций означает извлечение из очереди, `put` — добавление в очередь. Кроме того, у виртуальной машины есть 26 регистров, которые обозначаются буквами от 'a' до 'z'. Изначально все регистры хранят нулевое значение. В языке Quack существуют следующие команды (далее под α и β подразумеваются некие абстрактные временные переменные):

+	Сложение: <code>get α , get β , put $(\alpha + \beta) \bmod 65536$</code>
-	Вычитание: <code>get α , get β , put $(\alpha - \beta) \bmod 65536$</code>
*	Умножение: <code>get α , get β , put $(\alpha \cdot \beta) \bmod 65536$</code>
/	Целочисленное деление: <code>get α , get β , put $\alpha \div \beta$</code> (будем считать, что $\alpha \div 0 = 0$)
%	Взятие по модулю: <code>get α , get β , put $\alpha \bmod \beta$</code> (будем считать, что $\alpha \bmod 0 = 0$)
>[register]	Положить в регистр: <code>get α , установить значение [register] в α</code>
<[register]	Взять из регистра: <code>put значение [register]</code>
P	Напечатать: <code>get α , вывести α в стандартный поток вывода и перевести строку</code>
P[register]	Вывести значение регистра [register] в стандартный поток вывода и перевести строку
C	Вывести как символ: <code>get α , вывести символ с ASCII-кодом $\alpha \bmod 256$ в стандартный поток вывода</code>
C[register]	Вывести регистр как символ: вывести символ с ASCII-кодом $\alpha \bmod 256$ (где α — значение регистра [register]) в стандартный поток вывода
:[label]	Метка: эта строка программы имеет метку [label]
J[label]	Переход на строку с меткой [label]

Z[register][label]	Переход если 0: если значение регистра [register] равно нулю, выполнение программы продолжается с метки [label]
E[register1][register2][label]	Переход если равны: если значения регистров [register1] и [register2] равны, исполнение программы продолжается с метки [label]
G[register1][register2][label]	Переход если больше: если значение регистра [register1] больше, чем значение регистра [register2], исполнение программы продолжается с метки [label]
Q	Завершить работу программы. Работа также завершается, если выполнение доходит до конца программы
[number]	Просто число во входном файле — put это число

Формат входного файла

Входной файл содержит синтаксически корректную программу на языке Quack. Известно, что программа завершает работу не более чем за 10^5 шагов. Программа содержит не менее одной и не более 10^5 инструкций. **Метки имеют длину от 1 до 10 и состоят из цифр и латинских букв.**

Формат выходного файла

Выведите содержимое стандартного потока вывода виртуальной машины в выходной файл.

Решение

```
import queue
from edx_io import edx_io

registers = {}
queue = queue.Queue()
labels = {}
p = 0
q = False

def put(v):
    queue.put(v % 65536, False)

def get():
    return queue.get(False)

def jump(lbl):
    global p
    p = labels[lbl]

def do_command(cmd, io):
    c = cmd[0]
    if c == '+':
        put(get() + get())
    elif c == '-':
        a = get()
        put(a - get())
```

```

elif c == '*':
    put(get() * get())
elif c == '/':
    a = get()
    b = get()
    put(0 if b == 0 else a // b)
elif c == '%':
    a = get()
    b = get()
    put(0 if b == 0 else a % b)
elif c == '>':
    registers[cmd[1]] = get()
elif c == '<':
    put(registers[cmd[1]])
elif c == 'P':
    if len(cmd) == 1:
        io.writeln(get())
    else:
        io.writeln(registers[cmd[1]])
elif c == 'C':
    if len(cmd) == 1:
        io.write(chr(get() % 256))
    else:
        io.write(chr(registers[cmd[1]] % 256))
elif c == ':':
    labels[cmd[1:]] = p
elif c == 'J':
    jump(cmd[1:])
elif c == 'Z':
    if registers[cmd[1]] == 0:
        jump(cmd[2:])
elif c == 'E':
    if registers[cmd[1]] == registers[cmd[2]]:
        jump(cmd[3:])
elif c == 'G':
    if registers[cmd[1]] > registers[cmd[2]]:
        jump(cmd[3:])
elif c == 'Q':
    global q
    q = True
else:
    put(int(cmd))

```

```

with edx_io() as io:
    tokens = [t.decode('ascii') for t in io.all_tokens()]
    labels = {
        label[1:]: i
        for i, label in enumerate(tokens)
        if label[0] == ':'
    }

```

```

registers = {
    chr(letter): 0 for letter in range(ord('a'), ord('z') + 1)
}

lt = len(tokens)

while p < lt and not q:
    do_command(tokens[p], io)
    p += 1

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.812	28676096	1349803	213100
1	OK	0.062	11493376	69	5
2	OK	0.062	11526144	232	174
3	OK	0.062	11505664	3	0
4	OK	0.109	11534336	100	14
5	OK	0.359	11624448	56	48890
6	OK	0.218	11489280	67	25000
7	OK	0.218	11481088	67	25000
8	OK	0.250	11538432	55	25000
9	OK	0.093	11501568	461	60
10	OK	0.093	11563008	11235	21
11	OK	0.140	11726848	23748	42
12	OK	0.187	12439552	66906	8905
13	OK	0.078	11608064	7332	954
14	OK	0.078	11558912	4611	602
15	OK	0.140	11964416	37968	5424
16	OK	0.062	11481088	14	2
17	OK	0.046	11489280	70	10
18	OK	0.046	11399168	350	50
19	OK	0.062	11509760	1750	250
20	OK	0.078	11583488	8750	1250
21	OK	0.156	11984896	43750	6250

22	OK	0.468	14327808	218750	31250
23	OK	0.125	11931648	34606	4721
24	OK	0.687	22827008	683180	6
25	OK	0.687	23744512	683102	0
26	OK	0.234	28676096	1349803	0
27	OK	0.796	16777216	491572	210483
28	OK	0.781	16543744	491488	212066
29	OK	0.812	16424960	491600	212082
30	OK	0.781	16281600	491502	213100
31	OK	0.781	16535552	491416	211950
32	OK	0.796	17514496	491520	212621
33	OK	0.796	17432576	491317	209697
34	OK	0.796	16822272	491514	210896
35	OK	0.781	17321984	491557	212093