

Задача Н. Интерпретатор языка Quack (3 балла)

Имя входного файла: `quack.in`
Имя выходного файла: `quack.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Язык Quack — забавный язык, который фигурирует в задаче G с Internet Problem Solving Contest 2004 (<http://ipsc.ksp.sk/2004/real/problems/g.html>). В этой лабораторной работе 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 (`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`.

<code>+</code>	Сложение: <code>get x, get y, put (x+y) modulo 65536</code>
<code>-</code>	Вычитание: <code>get x, get y, put (x-y) modulo 65536</code> .
<code>*</code>	Умножение: <code>get x, get y, put (x*y) modulo 65536</code> .
<code>/</code>	Целочисленное деление: <code>get x, get y, put x div y</code> . (будем считать, что $0 \div 0 = 0$)
<code>%</code>	Взятие по модулю: <code>get x, get y, put x modulo y</code> . (будем считать, что $0 \bmod 0 = 0$)
<code>>[register]</code>	Положить в регистр: <code>get x, установить значение [register] в x</code> .
<code><[register]</code>	Взять из регистра: <code>put значение [register]</code> .
<code>P</code>	Напечатать: <code>get x, вывести x в стандартный поток вывода и перевести строку</code> .
<code>P[register]</code>	Вывести значение регистра <code>[register]</code> в стандартный поток вывода и перевести строку.
<code>C</code>	Вывести как символ: <code>get x, вывести символ с ASCII кодом x modulo 256 в стандартный поток вывода</code> .
<code>C[register]</code>	Вывести регистр как символ: вывести символ с ASCII кодом <code>x modulo 256</code> (где <code>x</code> — значение регистра <code>[register]</code>) в стандартный поток вывода.
<code>: [label]</code>	Метка: эта строка программы имеет метку <code>[label]</code> .
<code>J[label]</code>	Переход на строку с меткой <code>[label]</code> .
<code>Z[register][label]</code>	Переход если 0: если значение регистра <code>[register]</code> равно нулю, выполнение программы продолжается с метки <code>[label]</code> .
<code>E[register1][register2][label]</code>	Переход если равны: если значения регистров <code>[register1]</code> и <code>[register2]</code> равны, исполнение программы продолжается с метки <code>[label]</code> .
<code>G[register1][register2][label]</code>	Переход если больше: если значение регистра <code>[register1]</code> больше, чем значение регистра <code>[register2]</code> , исполнение программы продолжается с метки <code>[label]</code> .
<code>Q</code>	Завершить работу программы. Работа также завершается, если выполнение доходит до конца программы.
<code>[number]</code>	Просто число во входном файле — <code>put</code> это число.

Формат входного файла

Входной файл содержит корректную синтаксически программу на языке Quack. Известно, что программа завершает работу не более чем за 100 000 шагов.

Формат выходного файла

Выведите содержимого стандартного потока вывода виртуальной машины в выходной файл.

Пример

quack.in	quack.out
100 0 :start >a Zaend <a <a 1 + - >b <b Jstart :end P	5050

Второй пример подразумевает UNIX-переводы строки (один символ с кодом 10).

Лабораторная работа №4

Двоичный поиск, двоичное дерево поиска и сложные очереди.

quack.in	quack.out
58	58
49	49
10	10
62	62
97	97
10	10
80	80
97	97
10	10
90	90
97	97
50	50
10	10
60	60
97	97
10	10
74	74
49	49
10	10
58	58
50	50
10	10
48	48
10	10
58	58
51	51
10	10
62	62
97	97
10	10
90	90
97	97
52	52
10	10
67	67
97	97
10	10
74	74
51	51
10	10
58	58
52	52
10	10
0	0
:1	:1
>a	>a
Pa	Pa
Za2	Za2
<a	<a
J1	J1
:2	:2
0	0
:3	:3
>a	>a
Za4	Za4
Ca	Ca
J3	J3
:4	:4