

Problem A. 黑客技术

Input file: lock.in
Output file: lock.out
Time limit: 3s
Memory limit: 256MB

最近傲娇少女幽香很忙，于是让跳蚤国王出了一道题给 ZJOI2015，一时间人人自危。你秘密潜伏了下来想窃取跳蚤国王的试题。

终于有一天，趁跳蚤国王出去练习跳高，你偷偷打开了跳蚤国王的电脑——结果发现试题被 10 个电子密码锁锁住了！

任何事情都阻挡不了你进省队的决心！每个电子密码锁内部有 10 道检测工序，输入密码后，每通过一道检测工序密码锁就会输出一行“ok”。如果输出了 10 行“ok”那么就解开了该密码锁。

根据常识知，跳蚤国王一定是用高级计算机语言 fleaplus 写代码的，然后用 fleaplus 的编译器编译成汇编代码，再用汇编编译器编译为机器码。

显然你没有这 10 个电子密码锁的 fleaplus 源代码，通过反汇编，你获得了这 10 个电子密码锁的汇编代码。

跳蚤的程序开始运行时会开出长度为 2^{23} 的 32 位有符号整数数组 a ，元素分别为 $a_0, a_1, \dots, a_{2^{23}-1}$ ，并全部初始化为 0。

在跳蚤的汇编语言中，有三种量：“@ c ”，“\$ c ”，“# c ”（不含引号）。其中 c 是个整数且 $c \in [-2^{31}, 2^{31}]$ 。

- “@ c ” 表示整数常量 c 。
- “\$ c ” 表示 a_c 。
- “# c ” 表示 a_{a_c} 。

如果程序运行过程中试图访问的量会导致数组下标越界，那么就会产生 Runtime Error 错误并异常退出。

汇编代码由若干行组成，每行是一条语句（从 1 开始编号），程序会从第 1 条语句从上往下执行。

语句有下面这么几种：

- 赋值语句：= a b
 - a, b 是某个量。
 - 表示把 b 的值赋给 a 。
 - 如果 a 是个整数常量那么会产生 Runtime Error 错误并异常退出。
- 输入语句：getc c 和 geti c
 - c 是某个量。
 - getc c 表示读入一个字符，将该字符的 ASCII 码存入 c 中。

- `getc` 表示读到下一个非空白字符的位置，读入十进制表示的 32 位有符号整数，存入 `c` 中。（空白字符即 ASCII 码为 9,10,13,32 的字符）
- 如果 `c` 是个整数常量那么会产生 `Runtime Error` 错误并异常退出。如果读入整数失败则输出相关信息并异常退出。
- 输出语句: `putc c` 和 `puti c`
 - `c` 是某个量。
 - `putc c` 表示输出一个字符，`c` 是该字符的 ASCII 码。
 - `puti c` 表示输出整数 `c` 的十进制表示。
- 条件跳转语句: `if c t`
 - `c,t` 是某个量。
 - 表示如果 `c` 不是 0，那么跳转到第 `t` 条语句继续执行（即下一条要执行的语句变为第 `t` 条语句）；
 - 如果是 0，那么不进行任何操作，继续执行下一条语句。
- 运算语句: `o a b c`
 - `a,b,c` 是某个量，`o` 是某个运算符。
 - 该语句表示把 `a` 和 `b` 进行 `o` 的运算，把结果赋值给 `c`。
 - 如果 `o` 是 `+` 则表示加法，即 $a + b$ 。
 - 如果 `o` 是 `-` 则表示减法，即 $a - b$ 。
 - 如果 `o` 是 `*` 则表示乘法，即 $a \times b$ 。
 - 如果 `o` 是 `/` 则表示做除法然后下取整，即 $\lfloor \frac{a}{b} \rfloor$ 。
 - 如果 `o` 是 `<` 则表示 $a < b$ 则结果为 1，否则为 0。
 - 如果 `o` 是 `==` 则表示 $a = b$ 则结果为 1，否则为 0。
 - 如果 `c` 是个整数常量或者作除法时除数为 0 那么会产生 `Runtime Error` 错误并异常退出。

程序运行过程中如果试图执行第 `-1` 条语句那么程序正常结束，如果试图执行除此之外的不存在的语句那么会产生 `Runtime Error` 错误并异常退出。

如果执行的语句条数超过 10^7 那么会产生 `Time Limit Exceeded` 错误并异常退出。

现在，是时候展现你卓越的黑客技术了！

Input

本题为提交答案题。所有输入数据 `lock1.in~lock10.in` 已放置在选手目录下。

每个输入文件表示一个密码锁的汇编代码。

我们还提供了一个来自跳蚤国的模拟器，能直接执行汇编代码。

在终端中先切换到该试题的目录下（Windows 选手请使用 `cmd`）。

然后在终端中运行：`./run <code>`。其中 `<code>` 是你要执行的汇编代码的文件名。例如：`./run lock1.in`（Windows 选手请使用 `run_win32 lock1.in`）

模拟器将从终端中读入数据。如果你想从文件中读入，请使用 `<` 来指定。

例如：`./run lock1.in <lock1.out`

Output

针对给定的 10 个输入文件 `lock1.in~lock10.in`，你需要分别给出你的输出文件 `lock1.out~lock10.out`。

每个输出文件包含一些字符，表示你输入的密码。

Example

见 `lock0.in` 与 `lock0.out`。

该程序会读入一个整数，如果这个整数是 666 那么就会输出 10 行 “ok”。

Scoring

每个密码锁为一个测试点，每个测试点 10 分。

对于每个密码锁，我们会以你的输出文件作为输入运行密码锁。

如果程序异常退出则该测试点 0 分。

否则，我们会检查密码锁输出的每一行，设有 x 行为 “ok”。如果 $x \geq 10$ 则该测试点满分，否则该测试点得 x 分。

Postscript

你花了很长时间破解密码锁，最后也没有破解出来。最后硬着头皮去了 ZJOI2015，发现自己的事迹被写在了题面里。

Note

题目中提到的“32 位有符号整数”等价于 C/C++ 中的 `int`，Pascal 中的 `longint`，能表示 $[-2^{31}, 2^{31})$ 以内的整数。计算过程可能会溢出，比如： $2147483647 + 1 = -2147483648$ ， $100000 \times 100000 = 1410065408$ 。

另附 ASCII 字符对照表：

ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符
0	NUT	32	(space)	64	@	96	'
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	X	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL