

第十七届全国青少年信息学
(计算机) 奥林匹克竞赛
NOI 2000

选手试题

第一试

竞赛地点 中国 澳门

竞赛时间 2000/3/25

本次竞赛共分两试，本试卷为第一试。

● 分数分布

第一试共三道题，每道题最高得分 40 分。三道题最多为 120 分，凡交卷者另加 30 分，总分最高为 150 分。

● 文件命名

选手提交的可执行程序应以“.exe”为后缀，名字分别与每道试题的英文名字相同（该名字为一个不超过 8 个字母的英文单词，附于每题标题后的括号内），名字不分大小写。例如，如果题目的标题为“瓷片项链(Ring)”，选手提交的执行文件名应为“Ring.exe”，并最终拷贝到指定软盘的根目录下。

源文件的名字应尽量与编译后的可执行程序同名。

选手程序应从 INPUT.TXT 中读入输入数据，不需要进行判错处理。执行结果存放到 OUTPUT.TXT 中。输入输出文件均不需指定目录。

● 程序运行要求

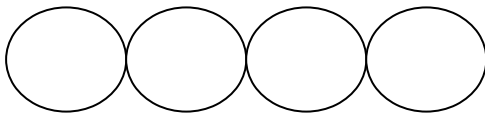
评测时采用黑盒测试，对每个测试点有一定的时间限制和分数。选手程序必须在此时限内运行结束并输出正确的结果才能得到这个点的分数。

程序运行时仅对指定的文件读写，不应在屏幕上显示任何信息或者等待键盘输入。输出结果应按题面要求，不应有多余的字符，测试系统会忽略选手输出中多余的空格和回车。

A. 瓷片项链(Ring)

原始部落用一种稀有的泥土烧制直径相同的圆瓷片并串成项链，串的时候沿瓷片的直径方向顺次连接，瓷片之间没有空隙也不重叠，一条项链至少由一个瓷片构成。

下图示出四片同样大小的瓷片串接所成的项链，其总长为单个瓷片直径的四倍。



每个烧制的瓷片厚度是一定的，直径 D 和所用泥土的体积 V 有以下关系：

$$D = \begin{cases} 0.3\sqrt{V - V_0} & (V > V_0) \\ 0 & (V \leq V_0) \end{cases}$$

其中 V_0 为烧制每一片的损耗，单位与 V 相同。当用料小于等于 V_0 时，不能烧制成瓷片。

例： $V_{\text{总}} = 10$ ， $V_0 = 1$ ，若烧制成一片瓷片， $V = V_{\text{总}} = 10$ ， $D = 0.9$ 。如果把泥土均分成 2 份，每份泥土的体积为 $V = V_{\text{总}}/2 = 5$ ，单个瓷片的直径为 $D' = 0.3\sqrt{5 - 1} = 0.6$ ，串起来的总长为 1.2。

给定了泥土的总体积和烧制单个瓷片的损耗，烧制的瓷片数不同，能够得到的项链总长度也不相同，请计算烧制多少个瓷片能使所得到的项链最长。

[输入文件]

文件仅有两行，每一行仅包含一个整数和一个换行/回车符。第一行的数字为泥土总体积 $V_{\text{总}}$ ($0 < V_{\text{总}} < 60000$)，第二行为烧制单个瓷片的损耗 V_0 ($0 < V_0 < 600$)。

[输出文件]

文件中仅包含一个数字和一个换行/回车符。该数字为能获得最长项链而烧制的瓷片数。如果不能烧制成瓷片或者最优解不唯一（存在两个或者两个以上方案均能获得最长项链），输出数字 0。

[输入输出文件样例 1]

[输入输出文件样例 2]

Input.txt
10
1
Output.txt
5

Input.txt
10
2
Output.txt
0

B. 程序分析器(Analyser)

Tiny Basm 语言(简称为 TB 语言)的巴科斯-瑙尔范式(BNF)为：

<程序> ::= <语句> ↓ { <语句> ↓ }
<语句> ::= <行号> □ <语句体>
<语句体> ::= <累加语句> | <输出语句> | <转移语句> | <条件语句> | <结束语句>
<累加语句> ::= <变量> + <整数>
<输出语句> ::= <变量> ?
<转移语句> ::= GO □ <行号>
<条件语句> ::= IF □ <变量> = <整数> □ <转移语句>
<结束语句> ::= END
<变量> ::= <字母>
<行号> ::= <整数>
<整数> ::= <数字> { <数字> }
<字母> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<数字> ::= 0|1|2|3|4|5|6|7|8|9

注：其中“::=”表示**定义为**，“|”表示**或**，{}内的项可以重复任意多次或不出现，“□”表示**空格**(一个字符，ASCII 码为 32)，“↓”表示**回车/换行**(两个字符，ASCII 码分别为 13 和 10)。

错误语句示例(在输入文件中不会出现任何错误语句)：

10□A+1.5 (不符合累加语句的定义，所加的不是整数)
20□A□? (不符合输出语句的定义，多加了一个空格)
30□IF□A=B□GO□10 (不符合条件语句的定义，不应变量=变量)

TB 程序的执行：

- 程序从行号最小的一条语句开始执行，在未遇到条件语句时按行号由小至大顺序执行。
- 所有变量在程序执行前被自动初始化为 0。
- 累加语句将语句中变量的值加上语句中的整数送回该变量。
- 输出语句将语句中变量的值在监视器上显示出来。
- 执行条件语句时，当且仅当该语句中的变量与紧跟在等号后面的整数值相等，后面的转移语句才被执行。该语句中的所有整数值至多为 4 位。
- 转移语句被执行后，程序将转去执行 GO 后面指定的行号的语句。
- 当程序执行结束语句后，结束整个程序的执行。
- 假设该系统能处理任意大小的整数，而不会发生溢出。

请编程，对于给定的 TB 语言程序 P，求该程序所执行的语句数(执行条件语句不论是否成功转移，仅记为执行一条语句)。

[输入文件]

- 输入文件为一个 TB 语言程序 P，语句数不超过 100 行。
- P 中每条语句的长度不超过 20 个字符。
- P 中转移语句里 GO 后面的行号一定有对应的语句。
- P 中可能有多个不同行号的结束语句。
- P 中行号最大的语句一定是结束语句。
- P 中的行号都不大于 3000。
- 输入文件不一定是按行号递增顺序给出 P 的。

[输出文件]

- 输出文件有且仅有一行：
如果程序能够正常结束，输出该程序所执行的语句数；
如果程序不能正常结束，输出-1。

[输入输出文件样例]

Input.txt

```
10 A+1
20 IF A=5 GO 60
60 END
30 A+2
40 A?
50 GO 20
```

Output.txt

```
11
```

[样例说明]

执行语句行号按顺序为

10→20→30→40→50→20→30→40→50→20→60

共 11 条语句被执行。

C. 古城之谜(LostCity)

著名的考古学家石教授在云梦高原上发现了一处古代城市遗址。让教授欣喜的是在这个他称为冰峰城(Ice-Peak City)的城市中有 12 块巨大石碑，上面刻着用某种文字书写的资料，他称这种文字为冰峰文。然而当教授试图再次找到冰峰城时，却屡屡无功而返。

幸好当时教授把石碑上的文字都拍摄了下来，为了解开冰峰城的秘密，教授和他的助手牛博士开始研究冰峰文，发现冰峰文只有陈述句这一种句型和名词(n)、动词(v)、辅词(a)这三类单词，且其文法很简单：

<文章> ::= <句子> { <句子> }

<句子> ::= <陈述句>

<陈述句> ::= <名词短语> { <动词短语> <名词短语> } [<动词短语>]

<名词短语> ::= <名词> | [<辅词>] <名词短语>

<动词短语> ::= <动词> | [<辅词>] <动词短语>

<单词> ::= <名词> | <动词> | <辅词>

注：其中<名词>、<动词>和<辅词>由词典给出，“::=”表示**定义为**，“|”表示**或**，{}内的项可以**重复任意多次或不出现**，[]内的项可以**出现一次或不出现**。

在研究了大量资料后，他们总结了一部冰峰文词典，由于冰峰文恰好有 26 个字母，为了研究方便，用字母 a 到 z 表示它们。

冰峰文在句子和句子之间以及单词和单词之间没有任何分隔符，因此划分单词和句子令石教授和牛博士感到非常麻烦，于是他们想到了使用计算机来帮助解决这个问题。假设你接受了这份工作，你的第一个任务是写一个程序，将一篇冰峰文文章划分为最少的句子在这个前提下，将文章划分为最少的单词。

[输入文件]

- 输入文件第 1 行为词典中的单词数 n ($n \leq 1000$)。
- 输入文件第 2 行至第 $(n+1)$ 行每行表示一个单词，形为“ $\alpha.mot$ ”， α 表示词性，可能是 n (名词), v (动词), a (辅词) 中的一个， mot 为单词，单词的长度不超过 20。拼写相同而词性不同的单词视为不同的单词，如输入示例中的 $n.kick$ 与 $v.kick$ 是两个不同的单词。
- 输入文件第 $(n+2)$ 行为需要划分的文章，以“.”结束。
- 输入文件中的文章确保为冰峰文。文章是由有限个句子组成的，每个句子只包含有限个单词。文章长度不超过 5KB。

[输出文件]

- 输出文件为两行，每行一个整数。
- 输出文件第 1 行为划分出来的句子数。
- 输出文件第 2 行为划分出来的单词数。

[输入输出文件样例]

Input.txt

```
11
n.table
n.baleine
a.silly
n.snoopy
n.sillysnoopy
v.is
v.isnot
n.kick
v.kick
a.big
v.cry
sillysnoopyisnotbigtablebaleinekickssnoopsillicry.
```

Output.txt

```
2
9
```

[样例说明]

(为了阅读方便，划分的单词用空格分隔，在单词的右上角标出它的词性，每行写一个句子，用句号表示句子结束。)

输出对应的划分：

sillysnoopyⁿ isnot^v big^a tableⁿ.

baleineⁿ kick^v snoopyⁿ silly^a cry^v.

如果用下面的划分：

silly^a snoopyⁿ isnot^v big^a tableⁿ.

baleineⁿ kick^v snoopyⁿ silly^a cry^v.

则划分的句子数仍为 2 个，但单词数却多了 1 个，为 10 个，显然应该按前者而不是后者划分。