

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

第二试

竞赛地点 中国 澳门

竞赛时间 2000/3/26

本次竞赛共分两试，本试卷为第二试。

● 分数分布

第二试共三道题，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。

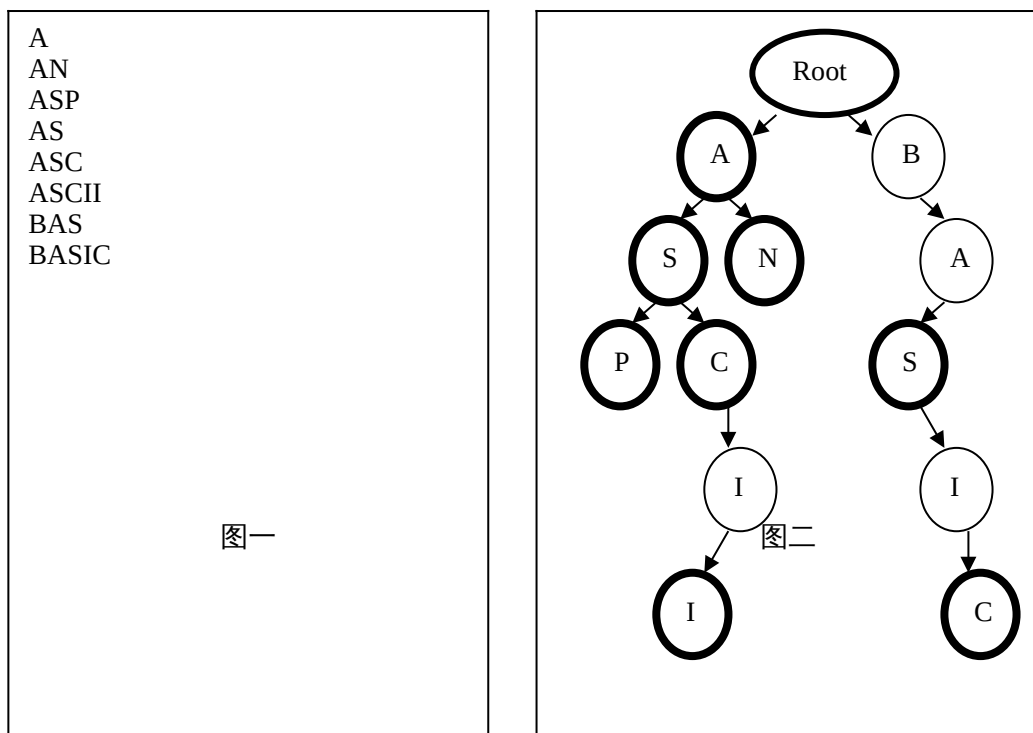
程序运行时仅对指定的文件读写，不应在屏幕上显示任何信息或者等待键盘输入。输出结果应按题面要求，不应有多余的字符，测试系统会忽略选手输出中多余的空格和回车。

D. 单词查找树(Tree)

在进行文法分析的时候，通常需要检测一个单词是否在我们的单词列表里。为了提高查找和定位的速度，通常都要画出与单词列表所对应的单词查找树，其特点如下：

- 根节点不包含字母，除根节点外每一个节点都仅包含一个大写英文字母；
- 从根节点到某一节点，路径上经过的字母依次连起来所构成的字母序列，称为该节点对应的单词。单词列表中的每个词，都是该单词查找树某个节点所对应的单词；
- 在满足上述条件下，该单词查找树的节点数最少。

例：图一的单词列表对应图二的单词查找树



对一个确定的单词列表，请统计对应的单词查找树的节点数（包括根节点）

[输入文件]

该文件为一个单词列表，每一行仅包含一个单词和一个换行/回车符。每个单词仅由大写的英文字符组成，长度不超过 63 个字符。文件总长度不超过 32K，至少有一行数据。

[输出文件]

该文件中仅包含一个整数和一个换行/回车符。该整数为单词列表对应的单词查找树的节点数。

[输入输出文件样例]

Input.txt

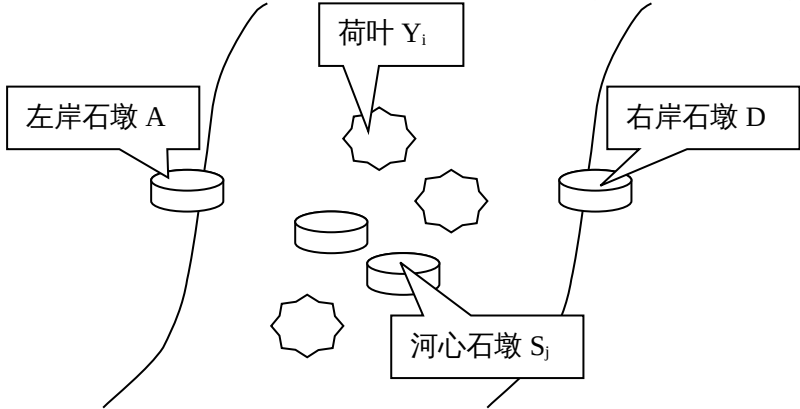
A
AN
ASP
AS
ASC
ASCII
BAS
BASIC

Output.txt

13

E. 青蛙过河(Frog)

大小各不相同的一队青蛙站在河左岸的石墩（记为 A）上，要过到对岸的石墩(记为 D)上去。河心有几片荷叶(分别记为 $Y_1 \dots Y_m$)和几个石墩(分别记为 $S_1 \dots S_n$)。图示如下：



青蛙的站队和移动方法规则如下：

1. 每只青蛙只能站在荷叶、石墩，或者**仅比它大一号**的青蛙背上（统称为合法的落脚点）；
2. 一只青蛙只有背上没有其它青蛙的时候才能够从一个落脚点跳到另一个落脚点；
3. 青蛙允许从左岸 A 直接跳到河心的石墩、荷叶和右岸的石墩 D 上，允许从河心的石墩和荷叶跳到右岸的石墩 D 上；
4. 青蛙在河心的石墩之间、荷叶之间以及石墩和荷叶之间可以来回跳动；
5. 青蛙在离开左岸石墩后，不能再返回左岸；到达右岸后，不能再跳回；
6. 假定石墩承重能力很大，允许无论多少只青蛙都可呆在上面。但是，由于石墩的面积不大，至多只能有一只青蛙直接站在上面，而其他的青蛙只能依规则 1 落在比它大一号的青蛙的背上。
7. 荷叶不仅面积不大，而且负重能力也有限，至多只能有一只青蛙站在上面。
8. 每一步只能移动一只青蛙，并且移动后需要满足站队规则；
9. 在一开始的时候，青蛙均站在 A 上，最大的一只青蛙直接站在石墩上，而其它的青蛙依规则 6 站在比其大一号的青蛙的背上。

青蛙希望最终能够全部移动到 D 上，并完成站队。

设河心有 m 片荷叶和 n 个石墩，请求出这队青蛙至多有多少只，在满足站队和移动规则的前提下，能从 A 过到 D。

例如，在 $m=1$ 且 $n=1$ 时，河心有一片荷叶 (Y_1) 和一个石墩 (S_1)，此时至多有 4 只青蛙能够过河（由小到大称为 1、2、3、4），过河的一种方法为：

开始		1 2 3 4 A S ₁ Y ₁ D
Step 1	1 from A to Y ₁	2 3 4 1 A S ₁ Y ₁ D
Step 2	2 from A to S ₁	3 4 2 1 A S ₁ Y ₁ D
Step 3	1 from Y ₁ to S ₁	3 1 4 2 A S ₁ Y ₁ D
Step 4	3 from A to Y ₁	1 4 2 3 A S ₁ Y ₁ D
Step 5	4 from A to D	1 2 3 4 A S ₁ Y ₁ D
Step 6	3 from Y ₁ to D	1 3 2 4 A S ₁ Y ₁ D
Step 7	1 from S ₁ to Y ₁	3 2 1 4 A S ₁ Y ₁ D
Step 8	2 from S ₁ to D	2 3 1 4 A S ₁ Y ₁ D
Step 9	1 from Y ₁ to D	1 2 3 4 A S ₁ Y ₁ D

此例中，当河心有一片荷叶和一个石墩时，4只青蛙能够跳动9步过河。

[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。第一行的数字为河心的石墩数 n ($0 \leq n \leq 25$)，第二行为荷叶数 m ($0 \leq m \leq 25$)。

[输出文件]

文件中仅包含一个数字和一个换行/回车符。该数字为在河心有 n 个石墩和 m 片荷叶时，最多能够过河的青蛙的只数。

[输入输出文件样例]

Input.txt

1
1

Output.txt

4

F. 算符破译(Equation)

考古学发现，几千年前古梅文明时期的数学非常的发达，他们懂得多位数的加法和乘法，其表达式和运算规则等都与现在**通常所用**的方式完全相同（如整数是十进制，左边是高位，最高位不能为零；表达式为中缀运算，先乘后加等），唯一的区别是其符号的写法与现在不同。有充分的证据表明，古梅文明的数学文字一共有 13 个符号，与 0,1,2,3,4,5,6,7,8,9,+,*,= 这 13 个数字和符号（称为现代算符）一一对应。为了便于标记，我们用 13 个小写英文字母 a,b,...m 代替这些符号（称为古梅算符）。但是，还没有人知道这些古梅算符和现代算符之间的具体对应关系。

在一个石壁上，考古学家发现了一组用古梅算符表示的等式，根据推断，每行有且仅有一个等号，等号左右两边为运算表达式（只含有数字和符号），并且等号两边的计算结果相等。

假设这组等式是成立的，请编程破译古梅算符和现代算符之间的对应关系。

[输入文件]

- 输入文件的第一行为等式的个数 N ($1 \leq N \leq 1000$)，以下 N 行每行为一个等式。
- 每个等式的长度为 5 个字符到 11 个字符。

[输出文件]

- 如果不存在对应关系能够满足这组等式，输出“noway”和一个换行/回车符。
- 如果有对应关系能够满足这组等式，输出所有能够确定的古梅算符和现代算符的对应关系。每一行有两个字符，其中第一个字符是古梅算符，第二个字符是对应的现代算符。输出按照字典顺序排序。

[输入输出文件样例]

Input.txt

```
2
abcdec
cdefe
```

Output.txt

```
a6
b*
d=
f+
```

[样例说明]

在上例中，可能对应的现代表达式为 $\{6*2=12, 2=1+1\}$ ， $\{6*4=24, 4=2+2\}$ ， $\{6*8=48, 8=4+4\}$ 。可见，能够确定的对应关系只有 a 对应 6，b 对应*，d 对应=，f 对应+，

应该输出；而{c,e}虽然能够找到对应的现代算符使得等式成立，但没有唯一的对应关系，不能输出。其他古梅算符{g,h...m}完全不能确定，也不能输出。