

2000 年全国青少年信息学（计算机）奥林匹克分区联赛复赛试题
(初中组 竞赛用时：3 小时)

题一：计算器的改良(18 分)

问题描述:

NCL 是一家专门从事计算器改良与升级的实验室，最近该实验室收到了某公司所委托的一个任务：需要在该公司某型号的计算器上加上解一元一次方程的功能。实验室将这个任务交给了一个刚进入的新手 ZL 先生。为了很好的完成这个任务,ZL 先生首先研究了一些一元一次方程的实例：

$$4+3x=8$$

$$6a-5+1=2-2a$$

$$-5+12Y=0$$

ZL 先生被主管告之，在计算器上键入的一个一元一次方程中，只包含整数、小写字母及十、一、=这三个数学符号（当然，符号“一”既可作减号，也可作负号）。方程中并没有括号，也没有除号，方程中的字母表示未知数。

问题求解：

编写程序，解输入的一元一次方程，将解方程的结果（精确至小数点后三位）输出至屏幕。

你可假设对键入的方程的正确性的判断是由另一个程序员在做，或者说可认为键入的一元一次方程均为合法的，且有唯一实数解。

样例

输入：

$$6a-5+1=2-2a$$

输出：

$$a=0.750$$

题二：税收与补贴问题(20 分)

问题描述：

每样商品的价格越低，其销量就会相应增大。现已知某种商品的成本及其在若干价位上的销量（产品不会低于成本销售），并假设相邻价位间销量的变化是线性的且在价格高于给定的最高价位后，销量以某固定数值递减。（我们假设价格及销售量都是整数）

对于某些特殊商品，不可能完全由市场去调节其价格。这时候就需要政府以税收或补贴的方式来控制。（所谓税收或补贴就是对于每个产品收取或给予生产厂家固定金额的货币）

问题求解：

你是某家咨询公司的项目经理，现在你已经知道政府对某种商品的预期价格，以及在各种价位上的销售情况。要求你确定政府对此商品是应收税还是补贴的最少金额（也为整数），才能使商家在这样一种政府预期的价格上，获取相对其他价位上的最大总利润。

总利润=单位商品利润*销量

单位商品利润=单位商品价格-单位商品成本（-税金 or +补贴）

输入：

输入的第一行为政府对某种商品的预期价，第二行有两个整数，第一个整数为商品成本，第二个整数为以成本价销售时的销售量，以下若干行每行都有两个整数，第一个为某价位时的单价，第二个为此时的销量，以一行-1，-1 表示所有已知价位及对应的销量输入完毕，输入的最后一行为一个单独的整数表示在已知的最高单价外每升高一块钱将减少的销量。

输出：

输出有两种情况：若在政府预期价上能得到最大总利润，则输出一个单独的整数，数的正负表示是补贴还是收税，数的大小表示补贴或收税的金额最小值。若有多解，取绝对值最小的输出。

如在政府预期价上不能得到最大总利润，则输出“NO SOLUTION”。

样例：

输入：

31

28 130

30 120

31 110

-1 -1

15

输出：

4

题三：乘积最大 (26 分)

问题描述:

今年是国际数学联盟确定的“2000——世界数学年”，又恰逢我国著名数学家华罗庚先生诞辰 90 周年。在华罗庚先生的家乡江苏金坛，组织了一场别开生面的数学智力竞赛的活动，你的一个好朋友 XZ 也有幸得以参加。活动中，主持人给所有参加活动的选手出了这样一道题目：

设有一个长度 N 的数字串，要求选手使用 K 个乘号将它分成 K+1 个部分，找出一种分法，使得这 K+1 个部分的乘积能够为最大。

同时，为了帮助选手能够正确理解题意，主持人还举了如下的一个例子：

有一个数字串: 312, 当 $N=3$, $K=1$ 时会有以下两种分法:

1) $3*12=36$

2) $31*2=62$

这时, 符合题目要求的结果是: $31*2=62$

现在, 请你帮助你的好朋友 XZ 设计一个程序, 求得正确的答案。

输入:

程序的输入共有两行:

第一行共有 2 个自然数 N, K ($6 \leq N \leq 40$, $1 \leq K \leq 6$)

第二行是一个 K 度为 N 的数字串。

输出:

结果显示在屏幕上, 相对于输入, 应输出所求得的最大乘积 (一个自然数)。

样例:

输入

4 2

1231

输出

62

题四: 单词接龙 (36 分)

问题描述:

单词接龙是一个与我们经常玩的成语接龙相类似的游戏, 现在我们已知一组单词, 且给定一个开头的字母, 要求出以这个字母开头的最长的“龙” (每个单词都最多在“龙”中出现两次), 在两个单词相连时, 其重合部分合为一部分, 例如 *beast* 和 *astonish*, 如果接成一条龙则变为 *beastonish*, 另外相邻的两部分不能存在包含关系, 例如 *at* 和 *atide* 间不能相连。

输入:

输入的第一行为一个单独的整数 n ($n \leq 20$) 表示单词数, 以下 n 行每行有一个单词, 输入的最后行为一个单个字符, 表来“龙”开头的字母。你可以假定以此字母开头的“龙”一定存在。

输出:

只需输出以此字母开头的最长的“龙”的长度

样例:

输入

5
at
touch
cheat
choose
tact
a

输出

23 (连成的“龙”为 atoucheatactactouchoose)

NOI 分区联赛 - 2000 年第六届普及组 试题解析

注意：解析和源程序均为 OIBH 站长刘汝佳所写，疏漏在所难免，但至少程序均通过了比赛时使用的测试数据，所以还是可以一看。

第一题：

比较简单。为了让程序编写时思路清晰和利于调试，可以采取如下的方法：等号两边分别处理，每次读一个“项”(先读符号，系数，再看是否有为一次项)，把方程简化成：

$ax+b=cx+d$ (假设 x 为未知数， a,b,c,d 为系数)

则 $x=(d-b)/(a-c)$

注意没有系数的情况，如： a 和 $1a$ 是等价的。大家注意看我的程序是如何处理的。

程序见附件。

第二题：

我先来说说题目的意思。就从样例开始分析。

输入是：

31
28 130
30 120
31 110
-1 -1
15

意思就是政府预期价是 31 元。成本 28 元，按成本销售的时候可以买 130 件产品。

每个卖 30 元的时候可以卖 120 个，

每个卖 31 元(输入的最高价位)的时候可以卖 110 个，
每个卖 32 元的时候可以卖：110-15=95 个。
每个卖 33 元的时候可以卖：110-15-15=80 个。
每个卖 34 元的时候可以卖：110-15-15-15=65 个。

...

因为“相邻价位之间的销量变化是均匀的”，因此 28 元卖 130 个,30 元卖 120 个
就可以知道

29 元卖 125 个(平均每元减少的销量是(130-120) div (30-28)=5)

输出是 4，我们来解释一下为什么是 4。

4 代表补贴是 4 元，所以：

在卖 28 元的时候，总利润是：(28-28+4)*130=520 元，

在卖 29 元的时候，总利润是：(29-28+4)*125=625 元，

在卖 30 元的时候，总利润是：(30-28+4)*120=720 元，

在卖 31 元的时候，总利润是：(31-28+4)*110=770 元，

在卖 32 元的时候，总利润是：(32-28+4)*95=760 元，

...

在卖 38 元的时候，总利润是：(38-28+4)*5=70 元，

显然可能的价位就是 28~38 了。(不能低于成本，卖 39 的时候销售量就是负数了)

可以看出，现在卖 31 元最划算，所以人们都愿意卖 31 元，这样一来不就达到政府的目的了吗！！

而当补贴是 0,1,2,3 的时候卖 31 元并不是最划算的，政府的目的达不到，你当然就没有分啦！

题意清楚了吗？好，下面分析思路。

穷举显然可以，但是没有什么意思，留给大家自己写。下面讲我的另外一种算法，数学味道要浓一些，希望大家坚持看完。

由于需要 N 元钱最划算，相当于使 N 元钱的利润大于等于每种价格的利润。因此可以分别考虑。

设补贴为 x，则 N 元钱的利润是：(p 为成本)

$$(N-p+x)*d[N]=(N-p)*d[N]+x*d[N]$$

因此 N 元钱比 M 元钱划算的时候有：

$$(N-p)*d[N]+x*d[N]>=(M-p)*d[M]+x*d[M], \text{即：}$$

$$x(d[N]-d[M])>=M*d[M]-N*d[N]-p*(d[M]-d[N])$$

这样，要使 N 元钱比 M 元钱划算，x 必须在区间[k1,k2] (k1,k2 根据上面的式子得出)

例如上面的例子：

31 元比 28 元划算时有：
 $(31-28+x)*110 \geq (28-28+x)*130$
即： $330+110x \geq 130x$, 故 $x \leq 16.5$

31 元比 30 元划算时有：
 $330+110x \geq 240+120x$, 故 $x \leq 9$

31 元比 32 元划算时有：
 $330+110x \geq 380+95x$, 故 $x \geq 3.33$

...
最后所有式子取交集，就得到了 x 的范围。要求绝对值最小值还不容易吗？ :-P
大家注意我在求出了 $k1, k2$ 后做的最后的处理。可能有一边或两边无界的情况。
正数和负数的处理也有区别。

有一点需要注意：题目没有说输入价位是从小到大排序好的，虽然测试数据都是排序好的。
我就偷个懒如何？ :-P

第三题：
搜索。数据很小，因此回溯就可以了。程序先预处理, 建立矩阵 $add[i, j]$
即第 j 个串连在第 i 个串之后能增加的长度。0 代表 j 不能增加 i 的后面。
一个需要注意的地方：
计算 $add[i, j]$ 的时候要计算最大可能值！
例如：
ABABABAB 和 ABABABC 就是 5, 不是 1！
现在没有问题了吧。为了方便和直观，我采用递归实现回溯搜索。程序见附件。

第四题：
有同学搜索第一个人，拣了以后第二个人用动态规划，一定能得最优解，但时间效率不大高。
有同学采用贪心，即用动态规划算出第一个人最大能拣的数，再在剩下的数中用动态规划。
反例如下：
1 9 1
0 0 0
1 9 1
第一次是：
1->9->9->1
第二次是：1
和是 21
但显然可以两次把所有的数拣完(22)。

本题是典型的多维动态规划，很象 IOI93 的第四题,另一个算法是网络流，很象 IOI97 第一题，
这里我只分析前者。这道题目的简单之处是阶段很好划分（对角线），这种方法我就不介绍了，
因为很多地方都有介绍。这里讲一种怪一点的动态规划^_^

容易想到的思路是：

令 $d[x_1, y_1, x_2, y_2]$ 表示甲在 x_1, y_1 , 乙在 x_2, y_2 以后（包括取走 (x_1, y_1) ）的过程中可以获得的最大和。

则 $d[1, 1, 1, 1]$ 就是原问题的解。

但是是否能取数和另一个人是否事先已经到过该格子有关，我们需要设计一种走的方法，使得只根据 x_1, y_1 , x_2, y_2 就能判断一些关键的格子是否已经到达过。这样，问题才具有无后效性。

为此，我们把路线作如下处理：

1) 当两个人路线有交叉的时候，改成等效的，不交叉的。

如下图，1 代表第一个人，2 代表第二个人。X 代表相遇点。

```
X111
2 1
222X2
  12
  12
  1X1
  2X
```

变成：

```
X222
1 2
111X2
  12
  12
  1X2
  1X
```

反正让 2 走右边就行了。

2) 现在 1 在第 y_1 列，2 在第 y_2 列，让 1 和 2 分别向右走，到达 yy_1 和 yy_2 列，然后向下走一格

这样如果 $yy_1 < yy_2$ ，便是分别取走第 $y_1 \sim yy_1, y_2 \sim yy_2$ 列数，否则路线有重复，就取走 $y_1 \sim yy_2$ 的数。

为了方便连续取数，我用了一个 $sum[x, y_1, y_2]$ 的数组，就是第 x 行的 $y_1 \sim y_2$ 的数。

请看我的程序中的相应部分。

这样，所有的走法都可以转换成上述的，具有无后效性的结构了。

由于递推是从 $d[x_1, y_1, x_2, y_2]$ 到 $d[x_1+1, y_1', x_2+1, y_2']$ ，而总有 $x_1=x_2=x$ ，所以可以把状态节

省为： $d[y_1, y_2]$ 而把 x (当前行) 作为阶段来递推：

```
for x:=n-1 downto 1 do
begin
  for y1:=1 to n do
    for y2:=y1 to n do
      枚举 y1' 和 y2' 作为新的 y1 和 y2, 注意保证 y1' >= y1, y2' >= y2 (题目规定),
      y1' <= y2' (刚才的分析), 递推 d[y1, y2]
      d1:=d2; {只记录相邻两个状态}
    end;
```

边界是什么呢？当然是从第 n 列的值了，就是 $\text{sum}[n, y_1, n]$ (从 y_1 取到 n)

一句话，就是两个人一起，一行一行往下走，每一行可以一次向右走若干步，但是要保证 2 在 1 的右边。

注意最后输出的是 $d_2[1, 1]$ 。如果输出 $d_1[1, 1]$, $n=1$ 会得到 0。（为什么？自己想啊）

附测试数据：<http://www.shzx.net.cn/cms/oi/shiti/2000fspdata.rar>

附源程序：<http://www.shzx.net.cn/cms/oi/shiti/2000fcpcode.rar>