第四届(1998 年)全国青少年信息学(计算机)奥林匹克分区联赛

(NOIP)高中组复赛试题（三小时完成）

题 1.火车从始发站(称为第 1 站)开出,在始发站上车的人数为 a,然后到达第 2 站,在第 2 站有人上、下车,但上、下车的人数相同，因此在第 2 站开出时（即在到达第 3 站之前）车上的人数保持为 a 人。从第 3 站起（包括第 3 站）上、下车的人数有一定的规律：上车的人数都是前两站上车人数之和，而下车人数等于上一站上车人数，一直到终点站的前一站（第 n-1 站），都满足此规律。现给出的条件是：共有 N 个车站，始发站上车的人数为 a,最后一站下车的人数是 m（全部下车）。试问从 x 站开出时车上的人数是多少？

输入：a,n,m 和 x

输出：x 站开出时车上的人数 (20%)

[分析]典型的数学题。为了找规律，我们建立一个表。

站号 1 2 3 4 5 6

开车时人数 num[] a a 2a 2a+b 3a+2b 4a+4b

上车人数 in[] a b a+b a+2b 2a+3b 3a+5b

下车人数 out[] 0 b b a+b a+2b 2a+3b

规律出来了，设第 k(k>=3)站时上车人数为 f[k-2]a+f[k-1]b (f[k]={1,1,2,3,5,8,13,21..}为 fibonacci 数列)容易证明，自己试一下吧。

num[k]=a+in[2]-out[2]+in[3]-out[3]...+in[k]-out[k]

而 in[2]=out[3],in[3]=out[4]...

故 num[k]=a-out[2]+in[k]=a-b+f[k-2]a+f[k-1]b= (f[k-2]+1)a + (f[k-1]-1)b (1)

因为知道第 n-1 站开车时人数为 m,容易求出 b,再代入(1)求第 x 站开车时的人数 p。即：

m=(f[n-3]+1)a + (f[n-2]-1)b (2)

p=(f[x-2]+1)a + (f[x-1]-1)b (3)

从(2)解得 b,代入(3)计算知

p=(f[x-2]+1)*a+(f[x-1]-1)*(m-(f[n-3]+1)*a) div (f[n-2]-1);

程序就只有 10 行了。注意 f[24]用 integer 装不下了，故只递推到 f[23]。

当然，你用枚举也可以，不过不如这种方法吸引人。

```
var f:array[1..23] of integer; i,a,n,m,x:integer;
begin f[1]:=1; f[2]:=1;
 for i:=3 to 23 do f[i]:=f[i-1]+f[i-2];
 write('a,n,m,x='); readln(a,n,m,x);
 writeln((f[x-2]+1)*a+(f[x-1]-1)*(m-(f[n-3]+1)*a) div (f[n-2]-1));
end.
```

题 2、设有 N 个正整数(N<=20), 将它们连接成一排, 组成一个最大的多位整数。

例如: N=3 时, 3 个整数 13,312,343 连接成的最大整数是: 34331213

例如: N=4 时, 4 个正整数 7,13,4,246 连接成的最大整数为: 7424613

程序输入: N N 个数

程序输出: 连接成的多位数

分析: 设有数组: VAR ST:ARRAY[1..20] OF STRING[20]; 该数组用于存放输入的 N 个正整数串, 再将这 N 个字串进行从大到小排序, 其 ASC 码大的放在前面, ASC 码小的放在后面, 最后从大到小依次输出字串, 就连接成了最大整数串.

若用数据测试将会发现有不对的时候, 例如 N=2, 2 个正整数 342 和 34, 因为'3'小于'2', 故'342'>'34', 输出结果为 34234. 但实际上 34234<34342, 即是说程序存在问题.

其实, 对于数组 ST 来说, 只需比较任意两个字串连接成的多位数, 即 ST[I]+ST[J]与 ST[J]+ST[I](交换连接). 若 ST[I]+ST[J]<ST[J]+ST[I]则 ST[I]与 ST[J]交换. 程序如下:

```
Type String10=String[10];
Var A:Array[1..20] Of Integer; St:Array[1..20] Of String10;
    S:String10; I,J,N:Integer;
Begin Write('N='); Readln(N);
If (N<1) Or (N>20) Then
 Begin Writeln('Input N Error!'); Halt; End; {输入错,程序中止}
```

```
For I:=1 To N Do Begin Read(A[I]);

Str(A[I],St[I]); End; Readln; {A[I]转成字串 ST[I]}

For I:=1 To N-1 Do For J:=I+1 To N Do {排序}

If St[I]+St[J]<St[J]+St[I] Then Begin S:=St[I]; St[I]:=St[J]; St[J]:=S End;

Write('Output:'); For I:=1 To N Do Write(St[I]); Writeln; {输出}

End.
```

题 3.(40%)著名科学家卢斯为了检查学生对进位制的理解,他给出了如下的一张加法表,表中的字母代表数字.(40%)

例如:

+	L	K	V	E
L	L	K	V	E
K	K	V	E	KL
V	V	E	KL	KK
E	E	KL	KK	KV

其含义:L+L=L,L+K=K,L+V=V,L+E=E,K+L=K,K+K=V,K+V=E,K+E=KL,...E+E=KV

根据这些规则可推导出:L=0,K=1,V=2,E=3

同时,可以确定该表表示的是 4 进制加法

程序输入:n(n<=9),表示行数,以下 N 行,每行 N 个字符串,每个字符串间用空格隔开.(字串仅有一个为'+'号,其他都由大写字母组成)

程序输出:(1)各个字母表示什么数,格式如:L=0,K=1,...

(2)加法运算是几进制的

(3)若不可能组成加法表,则应输出"ERROR!"

[分析]看起来很复杂，其实并不难。题目对“加法表”没有说清楚。虽然是取通常意义，题目也应该重新叙述。当年我就是把它理解的太宽，代码写了很多，虽然正确了，但浪费了不少时间。这道题说的“加法表”其实是狭义的，设为 k 进制，则第一行/第一列只能是 0,1,2,3..k-1.(我当时考虑了二位甚至更多位的情况),那么程序的第一步应该是：把输入数据储存在 T[0..8,0..8]中(最多 9 行嘛！),删除相同行和列(我很懒，这一步就算了吧,作为是 k*k 的矩阵.. :-D)来确定 k(剩下的不同的字母)。如果出现二位以上的数或第一行/列的元素不同，

则是"ERROR!"(我也懒得写了)然后匹配:若第一列为 M 的行有 L 个两位数，则 M=L(想一想，为什么),如果 M>=k 则"ERROR!"如果有两个数相等就"ERROR!"最后把得到的数代入加法表，验算一下，如果不对，就"ERROR!"

你也看到了，我有几处出错处理懒得写了，你自己加上吧，但是最后三处有代表性的我还是写了。程序如下。

```
var t:array[0..8,0..8] of string[2]; v:array['A'..'Z'] of integer;
 s:string; c,c2:char; i,j,n,p,k,tot:integer;
procedure error;
begin writeln('ERROR!'); halt;end;
begin for c:='A' to 'Z' do v[c]:=-1;
 write('N='); readln(n); k:=n-1;
 for i:=0 to n-1 do begin readln(s); s:=s+' '; c:=s[1];
 if i<>0 then v[c]:=0; for j:=0 to n-1 do
 begin while s[1]=' 'do delete(s,1,1);
          p:=pos(' ',s); t[i,j]:=copy(s,1,p-1);
 if p=3 then inc(v[c]); delete(s,1,p);
 end;
 end;
 tot:=0;
 for c:='A' to 'Z' do if v[c]<>-1 then inc(tot);
 if tot<>k then error;
 for c:='A' to 'Y' do for c2:=succ(c) to 'Z' do
 if (v[c]<>-1)and(v[c]=v[c2]) then error; {equal}
 {check}
 for i:=1 to k do for j:=1 to k do
```

```
begin if length(t[i,j])=2 then p:=v[t[i,j,1]]*k+v[t[i,j,2]]
else p:=v[t[i,j,1]]; if v[t[i,0,1]]+v[t[0,j,1]]<>p then error;
end;
for c:='A' to 'Z' do if v[c]<>-1 then write(c,'=',v[c],' ');
writeln; writeln('RADIX=',k);
end.
```

附：测试数据：http://www.shzx.net.cn/cms/oi/shiti/1998fstdata.zip

刘汝佳的分析和源程序：http://www.shzx.net.cn/cms/oi/shiti/NOIP98_T_F.rar