2002 年第八届全国青少年信息学（计算机）奥林匹克分区联赛复赛试题
（普及组 竞赛用时：3 小时）

题一 级数求和（存盘名：NOIPC1）

[问题描述]:

已知：Sn= 1＋1／2＋1／3＋…＋1／n。显然对于任意一个整数 K，当 n 足够大的时候，Sn 大于 K。

现给出一个整数 K（1<=k<=15），要求计算出一个最小的 n；使得 Sn＞K。

[输入]

键盘输入 k

[输出]

屏幕输出 n

[输入输出样例]

输人：1

输出：2

题二 选数（存盘名：NOIPC2）

[问题描述]:

已知 n 个整数 x1,x2,…,xn，以及一个整数 k（k＜n）。从 n 个整数中任选 k 个整数相加，可分别得到一系列的和。例如当 n=4，k＝3，4 个整数分别为 3，7，12，19 时，可得全部的组合与它们的和为：

3＋7＋12=22　　3＋7＋19＝29　　7＋12＋19＝38　　3＋12＋19＝34。

现在，要求你计算出和为素数共有多少种。

例如上例，只有一种的和为素数：3＋7＋19＝29）。

[输入]：

键盘输入，格式为：

n，k （1<=n<=20，k＜n）

x1,x2，…,xn （1<=xi<=5000000）

[输出]：

屏幕输出，格式为：

一个整数（满足条件的种数）。

[输入输出样例]:

输入：

4 3

3 7 12 19

输出：

1

题三 产生数（存盘名：NOIPC3）

[问题描述]:

给出一个整数 n（n<10^30) 和 k 个变换规则（k<=15）。

规则：

一位数可变换成另一个一位数：

规则的右部不能为零。

例如：n=234。有规则（k＝2）：

2－> 5

3－> 6

上面的整数 234 经过变换后可能产生出的整数为（包括原数）:

234

534

264

564

共 4 种不同的产生数

问题：

给出一个整数 n 和 k 个规则。

求出：

经过任意次的变换（0 次或多次），能产生出多少个不同整数。

仅要求输出个数。

[输入]:

键盘输人，格式为：

n k

x1 y1

x2 y2

... ...

xn yn

[输出]:

屏幕输出，格式为：

一个整数（满足条件的个数）：

[输入输出样例]:

输入:

234 2

2 5

3 6

输出：

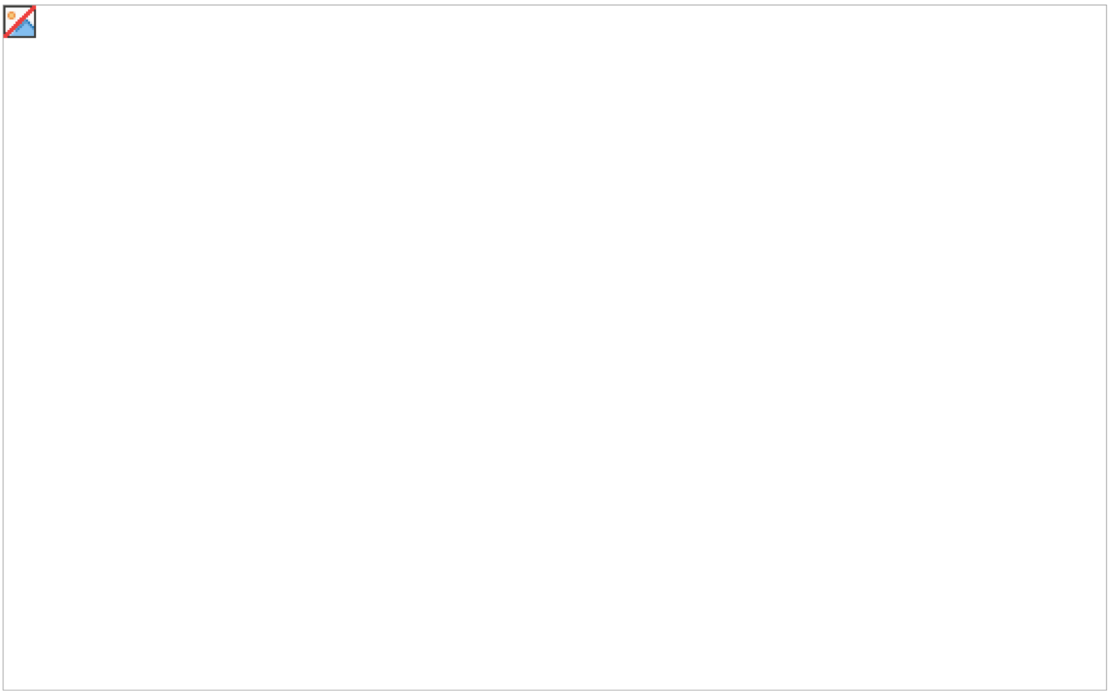
4

题四 过河卒（存盘名：NOIPC4）

[问题描述]:

如图，A 点有一个过河卒，需要走到目标 B 点。卒行走规则：可以向下、或者向右。

同时在棋盘上的任一点有一个对方的马（如上图的 C 点），该马所在的点和所有跳跃一步可达的点称为对方马的控制点。例如上图 C 点上的马可以控制 9 个点（图中的 P1，P2 … P8 和 C）。卒不能通过对方马的控制点。



　　棋盘用坐标表示，A 点（0，0）、B 点（n,m）(n,m 为不超过 20 的整数，并由键盘输入)，同样马的位置坐标是需要给出的（约定: C<>A，同时 C<>B）。现在要求你计算出卒从 A 点能够到达 B 点的路径的条数。

[输入]:

　　键盘输入

　　　B 点的坐标（n,m）以及对方马的坐标（X,Y）{不用盘错}

[输出]:

　　屏幕输出

　　　　一个整数（路径的条数）。

[输入输出样例]:

　　输入：

　　　6 6 3 2

　　输出：

　　　17

NOIP2002 普及组试题解题报告

天津市南大附中 耿昊炎

第一题 级数求和

[问题描述]：

已知：Sn=1+1/2+1/3+…+1/n，显然对于任意一个整数 K，当 n 足够大的时候，Sn 大于 K。现给出一个整数 K (1≤K≤15)，要求计算出一个最小的 n，使得 Sn>K。

[输入]：

键盘输入 k

[输出]：

屏幕输出 n

[输入输出样例]：

输入： 1

输出： 2

算法分析：由于本题给出的 K 的范围较小(1≤K≤15)，所以采用穷举算法，即每当 n 值增加 1，就算出当前 1/n 的值，存于变量 a 中，变量 s 作为累加器，即 Sn 的值。当 s>K 时，停止循环，输出当前的 n 值，即为所求。

考虑到精确度问题，本程序运算过程中采用双精度(Double)数据类型。

程序如下：

```
CLS
DEFDBL A, N, S
INPUT k
DO
n = n + 1
a = 1 / n
s = s + a
LOOP UNTIL s > k
PRINT i
END
```

小结：本题的算法，只适用于 K 值较小的情况。当 K 值很大时，就要专门编写高精度加法、除法的子程序，或设计其他算法，在此就不做讨论了。

第二题 选 数

[问题描述]：

已知 n 个整数 x1,x2,…,xn，以及一个整数 k(k<n)。从 n 个整数中任选 k 个整数相加，可分别得到一系列的和。例如当 n=4,k=3，4 个整数分别为 3，7，12，19 时，可得全部的组合与它们的和为：

3+7+12=22　　3+7+19=29　　7+12+19=38　　3+12+19=34。

现在，要求你计算出和为素数共有多少种。

例如上例，只有一种的和为素数：(3+7+19=29)。

[输入]：键盘输入，格式为：

n,k (1≤n≤20，k<n)

x1,x2,…xn (1≤xi≤5000000)

[输出]：屏幕输出 ，格式为：

一个整数(满足条件的种数)。

[输入输出样例]：

输入：

4 3

3 7 12 19

输出：

1

算法分析：从 n 个数中选 k 个，如果采用搜索，总搜索次数为 。从题目的规模看，n 最大为 20，k<n。所以，采用搜索最多要进行 =184756 次，在时间上可以承受。所以本题采用递归搜索算法。并且，因为 1≤xi≤5000000，所以求和过程中可能得到的最大值为 20*5000000=100,000,000，用双精度数完全可以解决，无需进行高精度计算。

本题中输入样例的格式为每两个数之间用空格隔开。这对使用 PASCAL 的同学没有影响，但使用 Quick BASIC 的同学要想使输入格式与题目要求相符，就要应用字符串截取。

程序中，主程序主要对输入进行处理，"make"是组合的产生过程，用递归的方法累加出不同的和。函数"ok"用于判断产生的和是否为素数。全局变量 s 统计总数，数组 a(n)存放 x1,x2,…,xn，used(i)标记在产生过程中数 a(i)是否已使用。

程序如下：

```
DECLARE SUB make (k#, a#, b#)
DECLARE FUNCTION ok# (a#)
CLS : DEFDBL A-Z: DIM SHARED n, s
INPUT inp$ '对输入数据的处理，到 NEXT i 为止
f = INSTR(inp$, " ")
n = VAL(LEFT$(inp$, f - 1)): k = VAL(RIGHT$(inp$, LEN(inp$) - f))
DIM SHARED a(n), used(n)
INPUT inp$: inp$ = inp$ + " "
FOR i = 1 TO n
f = INSTR(inp$, " ")
a(i) = VAL(LEFT$(inp$, f - 1))
inp$ = RIGHT$(inp$, LEN(inp$) - f)
NEXT i
CALL make(k, 0, 1)
PRINT s
END
SUB make (k, a, b)
IF k = 0 THEN
IF ok(a) THEN s = s + 1
EXIT SUB
END IF
FOR i = b TO n
IF used(i) = 0 AND k > 0 THEN
used(i) = 1
CALL make(k - 1, a + a(i), i + 1)
used(i) = 0
END IF
NEXT i
END SUB
FUNCTION ok (a)
ok = 1
```

```
IF a / 2 = a \ 2 THEN
ok = 0
ELSE
FOR i = 3 TO SQR(a) + 1
IF a \ i = a / i THEN ok = 0: EXIT FOR
NEXT i
END IF
END FUNCTION
```

第三题 产生数

[问题描述]：

给出一个整数 n(n<1030)和 k 个变换规则(k≤15)。

规则：

1 位数可变换成另一个一位数；

规则的右部不能为零。

例如：n=234，有规则(k=2)：

2 → 5

3 → 6

上面的整数 234 经过变换后可能产生出的整数为(包括原数)：

234

534

264

564

共 4 种不同的产生数。

问题：

给出一个整数 n 和 k 个规则。

求出：

经过任意次的变换(0 次或多次)，能产生出多少个不同整数。仅要求输出个数。

[输入]：键盘输入，格式为：

n k

x1 y1

x2 y2

…

xn yn

[输出]：屏幕输出，格式为：

一个整数(满足条件的个数)。

[输入输出样例]：

输入：

234 2

2 5

3 6

输出：

4

算法分析：本题采用搜索算法。

由于输入数据规模达到 1030，n 的长度最大可达 31 位，所以输入数据和对整数 n 的处理要用字符串。另

外，因为 k 最大可达到 15，所以对于极限数据，每一位数都有可能变换成为其它任意一个 1 位整数。如果采用对整个字串进行搜索的方法，最大搜索次数就将达到 1030 次。而且本题还要有判重的处理(产生相同的数视为一种)，显然在时间和空间上都将令人无法忍受。但是仔细观察可以发现，本题数的变换仅局限于每一位，就是说每一位数的变换总数只与自己有关，和其它数位没有关系。于是可以用分治法先求出每一位数能产生出多少个不同整数，再运用乘法原理，求得总数 num = 。

上面已经提到，本题的数据已经远远超过了双精度数的范围，所以还须在统计总数时应用高精度计算。

另外，同第二题一样，本题输入数据的格式必须用字符串查找进行处理。

程序中，主程序主要对输入进行处理和计算总数。过程"Search"分别对每一位进行搜索，数组 s(l)储存每一位上的搜索结果。函数"ok"及数组 re$(10)对每次产生的数进行判重。

程序如下：

```
DECLARE SUB search (a$)
DECLARE FUNCTION ok! (x$)
CLS : DIM SHARED k, l, p, re$(10), s1(100)
INPUT inp$: f = INSTR(inp$, " ")
n$ = LEFT$(inp$, f - 1): k = VAL(RIGHT$(inp$, LEN(inp$) - f)): l = LEN(n$)
DIM SHARED a$(k, 2), s(l)
FOR i = 1 TO k
INPUT inp$: f = INSTR(inp$, " ")
a$(i, 1) = LEFT$(inp$, f - 1): a$(i, 2) = RIGHT$(inp$, LEN(inp$) - f)
NEXT i
FOR i = 1 TO l
p = 0: CALL search(MID$(n$, i, 1))
s(i) = p
NEXT i
s1(100) = 1: k = 100
FOR j = 1 TO l
FOR i = 100 TO k STEP -1
s1(i) = s1(i) * s(j)
NEXT i
FOR i = 100 TO k STEP -1
IF s1(i) > 9 THEN
s1(i - 1) = s1(i - 1) + s1(i) \ 10
s1(i) = s1(i) MOD 10
IF i = k THEN k = k - 1
END IF
NEXT i
NEXT j
FOR i = k TO 100: PRINT LTRIM$(STR$(s1(i))); : NEXT i
END
FUNCTION ok (x$)
ok = 1
FOR i = 1 TO p
IF x$ = re$(i) THEN ok = 0: EXIT FOR
NEXT i
```

```
END FUNCTION
SUB search (a$)
IF ok(a$) = 0 THEN EXIT SUB ELSE p = p + 1: re$(p) = a$
FOR i = 1 TO k
IF a$ = a$(i, 1) THEN
CALL search(a$(i, 2))
END IF
NEXT i
END SUB
```

第四题 过河卒

[问题描述]：

如图：A 点有一个过河卒，需要走到目标 B 点。卒行走的规则：可以向下，或者向右。同时在棋盘上的任一点有一个对方的马(如上图的 C 点)，该马所在的点和所有跳跃一步可达到的点称为对方马的控制点。例如上图 C 点上的马可以控制 9 个点(图中的 P1,P2……P8 和 C)。卒不能通过对方马的控制点。

棋盘用坐标表示，A 点(0,0)、B 点(n,m) (n,m 为不超过 20 的整数，并由键盘输入)，同样马的位置坐标是需要给出的(约定：C≠A，同时 C≠B)。现在要求你计算出卒从 A 点能够到达 B 点的路径的条数。

[输入]：

键盘输入

B 点的坐标(n,m)以及对方马的坐标(X,Y) {不用判错}

[输出]：

屏幕输出

一个整数(路径的条数)。

[输入输出样例]：

输入：

6 6 3 2

输出：

17

算法分析：本题采用递推算法。

设当前卒的位置为(i,j)，a(i,j)为从(0,0)到当前位置的总路径数，则：

a(i,j) = a(i-1,j) + a(i,j-1) (*)

由此，利用(*)式为递推公式，求 a(i,j)可转化为求 a(i-1,j)与 a(i,j-1)，最后转化到 a(0,0)，而 a(0,0)=1。这样就可以推出 a(x,y)，即问题所求。

值得注意的是，本题图中共有九个点是卒无法到达的。所以，需要函数"ok"及数组 d(8,2)来判断当前位置是否可以到达。如果可以，则进行计算，否则跳过该点。

本题的数据规模不是很大，但本人为了保险起见，在竞赛中对总数处理采用了高精度计算，经试验，至少 n=m=150 时都可顺利出解。后来实践证明，按照本题的规模，用双精度就可以解决。

另外，为了使输入处理这一段程序更加简洁，n,m,x,y 的值统一存入数组 i(4)中。

程序如下：

```
DECLARE FUNCTION ok! (x1!, y1!)
CLS : DIM SHARED i(4)
INPUT inp$: inp$ = inp$ + " "
FOR i = 1 TO 4
f = INSTR(inp$, " ")
```

```
i(i) = VAL(LEFT$(inp$, f - 1))
inp$ = RIGHT$(inp$, LEN(inp$) - f)
NEXT i
DATA -2,1,-1,2,1,2,2,1,2,-1,1,-2,-1,-2,-2,-1: DIM SHARED a(100, -1 TO i(2)), d(8, 2)
FOR i = 1 TO 8: READ d(i, 1), d(i, 2): d(i, 1) = d(i, 1) + i(3): d(i, 2) = d(i, 2) + i(4): NEXT i
a(100, 0) = 1: a = 100
FOR i = 0 TO i(1)
FOR j = 0 TO i(2)
IF ok(i, j) THEN
FOR k = 100 TO a STEP -1
a(k, j) = a(k, j) + a(k, j - 1)
IF a(k, j) > 9 THEN
a(k - 1, j) = a(k - 1, j) + a(k, j) \ 10
a(k, j) = a(k, j) MOD 10
IF k = a THEN a = a - 1
END IF
NEXT k
ELSE
FOR k = 100 TO a STEP -1
a(k, j) = 0
NEXT k
END IF
NEXT j
NEXT i
FOR i = 1 TO 100
IF a(i, i(2)) <> 0 THEN p = 1
IF p = 1 THEN PRINT CHR$(a(i, i(2)) + 48);
NEXT i
END
FUNCTION ok (x1, y1)
ok = 1
IF x1 = i(3) AND y1 = i(4) THEN
ok = 0
ELSE
FOR i = 1 TO 8
IF x1 = d(i, 1) AND y1 = d(i, 2) THEN ok = 0: EXIT FOR
NEXT i
END IF
END FUNCTION
```

附测试源程序：http://www.shzx.net.cn/cms/oi/shiti/2002cppcode.rar

附测试数据：http://www.shzx.net.cn/cms/oi/shiti/2002fspdata.zip

附解题报告(二)湖南 黄艺海：http://www.shzx.net.cn/cms/oi/shiti/NOIp2002p_Report_Diablo.rar