

时限：1s

空限：256M

## 1. 产生数(build.cpp)

### 【问题描述】

给出一个整数  $n$  ( $n < 10^{30}$ ) 和  $m$  个变换规则 ( $m \leq 20$ )。

约定：一个数字可以变换成另一个数字，规则的右部不能为零，即零不能由另一个数字变换而成。而这里所说的一个数字就是指一个一位数。

现在给出一个整数  $n$  和  $m$  个规则，要你求出对  $n$  的每一位数字经过任意次的变换 (0 次或多次)，能产生出多少个不同的整数。

### 【输入】

共  $m+2$  行，第一行是一个不超过 30 位的整数  $n$ ，第二行是一个正整数  $m$ ，接下来的  $m$  行是  $m$  个变换规则，每一规则是两个数字  $x$ 、 $y$ ，中间用一个空格间隔，表示  $x$  可以变换成  $y$ 。

### 【输出】

仅一行，表示可以产生的不同整数的个数。

### 【样例】

build.in

123

2

1 2

2 3

build.out

6

## 2. 青蛙过河(frog.cpp)

### 【问题描述】

有一条河，左边一个石墩（A 区）上有编号为 1, 2, 3, 4, ..., n 的 n 只青蛙，河中有 k 个荷叶（C 区），还有 h 个石墩（D 区），右边有一个石墩（B 区），如下图 2-5 所示。n 只青蛙要过河（从左岸石墩 A 到右岸石墩 B），规则为：

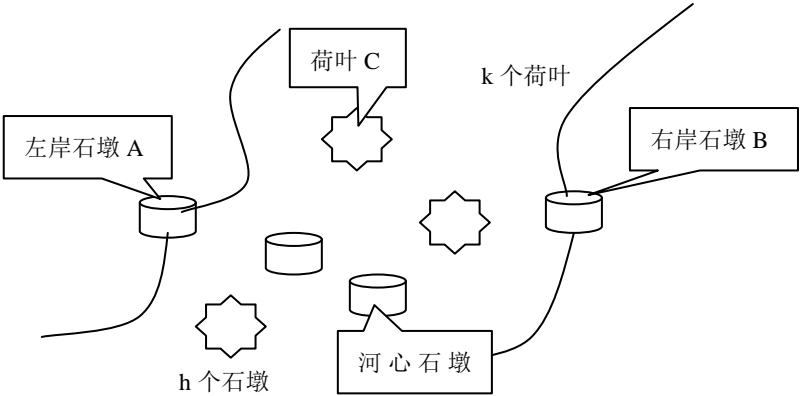


图 2-5

- (1) 石墩上可以承受任意多只青蛙，荷叶只能承受一只青蛙（不论大小）；
- (2) 青蛙可以： $A \rightarrow B$ （表示可以从 A 跳到 B，下同）， $A \rightarrow C$ ， $A \rightarrow D$ ， $C \rightarrow B$ ， $D \rightarrow B$ ， $D \rightarrow C$ ， $C \rightarrow D$ ；
- (3) 当一个石墩上有多只青蛙时，则上面的青蛙只能跳到比它大 1 号的青蛙上面。

### 【你的任务】

给出 h,k ( $0 \leq h, k \leq 25$ )，输出最多能有多少只青蛙可以根据以上规则顺利过河？

### 【样例】

```
frog.in
2 3      {河中间 2 个石墩， 3 个荷叶}
frog.out
16       {最多有 16 只青蛙可以按照规则过河}
```

### 3. 最大乘积 (maxmul.cpp)

#### 【问题描述】

一个正整数一般可以分为几个互不相同的自然数的和，如  $3=2+1$ ， $4=1+3$ ， $5=1+4=2+3$ ， $6=1+5=2+4$ ， $\dots$ 。

现在你的任务是将指定的正整数  $n$  分解成若干个互不相同的自然数的和，且使这些自然数的乘积最大。

#### 【输入】

只一个正整数  $n$  ( $3 \leq n \leq 10000$ )。

#### 【输出】

第一行是分解方案，相邻的数之间用一个空格分开，并且按由小到大的顺序。

第二行是最大的乘积。

#### 【样例】

maxmul.in

10

maxmul.out

2 3 5

30

## 4. 算 24 点(point. cpp)

### 【问题描述】

几十年前全世界就流行一种数字游戏，至今仍有人乐此不疲。在中国我们把这种游戏成为“算 24 点”。您作为游戏者得到 4 个 1~9 之间的自然数作为操作数，而您的任务是对这 4 个操作数进行适当的算术运算，要求运算结果等于 24。

您可以使用的运算只有：+，-，\*，/，您还可以使用 () 来改变运算顺序。注意：所有的中间结果必须是整数，所以一些除法运算是允许的（例如， $(2*2)/4$  是合法的， $2*(2/4)$  是不合法的）。下面我们给出一个游戏的具体例子：

若给出的 4 个操作数是：1、2、3、7，则一种可能的解答是  $1+2+3*7=24$ 。

### 【输入】

只有一行，四个 1 到 9 之间的自然数。

### 【输出】

如果有解的话，只要输出一个解，输出的是三行数据，分别表示运算的步骤。

题目要求：1、题中两个运算符是不允许相邻的，例如  $5+ -3=2$  是错误的。

2、在第一次运算后，优先考虑另两个数的运算。例如 2 2 2 3

前两个  $2+2=4$  运算后，先考虑  $2*3=6$  而不是  $4*2=8$ 。

3、要使最后结果为 24。4 个操作数的顺序不定。

4、如果没有解则输出 “No answer!”

5、一个运算式中输出顺序为大数+运算符+小数。

### 【样例】

point.in  
1 2 3 7

point.out  
 $2+1=3$   
 $7*3=21$   
 $21+3=24$

