证明计算机二进制数的 1 的个数的程序：

f(x)= if x==0 then 0

else then x&1+f(x/2)

其中 x 是大于 0 的整数，要证明对于所有正整数，f(x)=x 表示为二进制数的 1 的个数

程序规范如下：

P(s): x>0

Q(x,z): z= x 表示为二进制数的 1 的个数

证明：

1)当 x=1 时，根据程序 f(1)=1+f(0)=1，正确

2)假设对任意的正整数 x，f(x)= x 表示为二进制数的 1 的个数成立，则对于 x+1 有：

f(x+1)=(x+1)&1+f((x+1)/2)

1. 如果 x 是偶数，则 f(x+1)=1+f((x+1)/2)=f(x)+1，即奇数的最后一位一定是 1，所以二进制表示的 1 的总数等于 1 加上去除最后一位后的数的二进制 1 的个数，符合 f(x)的含义。

2. 如果 x 是奇数，则 f(x+1)=f(x/2)，即偶数的最后一位一定不包含 1，所以可以递归处理去除最后一位后的数，符合 f(x)的定义。

综上所述，证明了对任何满足 P(x)的 x，程序 f(x)执行始终是正确的。

程序的终止性：取良序集(N,<)，归纳总会在有限步结束，否则与良序集(N,<)相矛盾:

证明：设命题 P(x_1)为 f(x_1)计算终止，则

(1) 对于 N 中的最小元素 0：f(0)=0，计算终止

(2) 若 $x_1 \neq 0$ f(x_1)=x$\&_1$+f(x_1/2)

∵$x_1/2<x_1$

∴根据假设，f(x_1)的计算会终止

转换为尾递归：

b(x)= x==0

h(x)= 0

F(x,y)=x+y

k(x)=x/2 g(x)=x&1

取良序关系为通常的小于关系<

∵x/2<x

∴k(x)<x

∵F(x,y)=x+y 满足结合律，F 的右单元为 0

b,h,g,k 中不含有 F，因此满足 Cooper 变换规则的可用性条件，有输出模式：

f(x)=G(x,0)

G(x,y)= if x==0 then y else G(x/2,x&1+y)

转换为非递归程序：

```
N(x):
   count:=0
while x!=0 do
      count:=count+x&1
      x:=x/2
z<-count
```