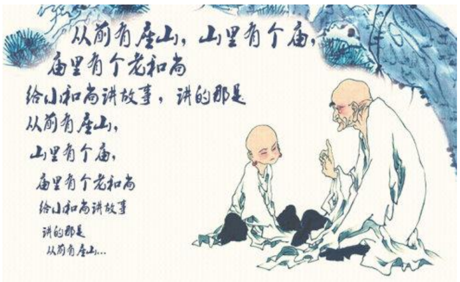
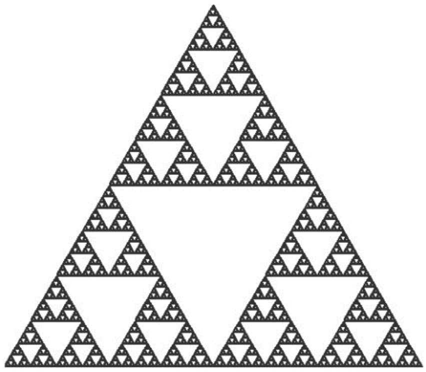
7.递　归 2

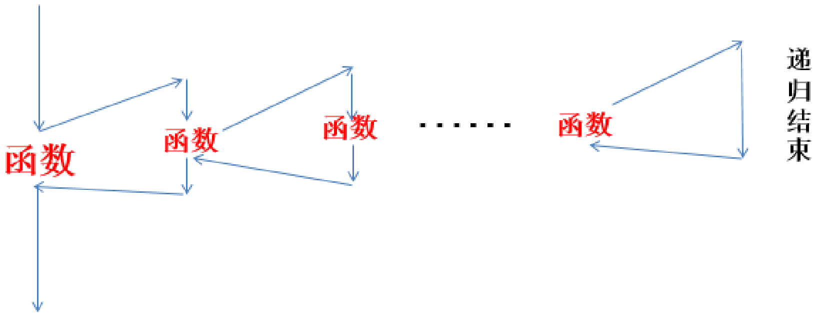
目录

一.递归的定义

儿子：“爸爸这个题怎么做？”
爸爸：“问你妈妈去！”
儿子：“妈妈这个题怎么做？”
妈妈：“问你爸爸去！”
…

上述现象称为递归。可能到某个时候停下来，也可能永远不停的进行下去…。
直接或间接的调用自己的函数，称为递归函数。递归函数必须有终止条件，否则不停的进行下去，造成栈溢出。



递归的关键：
1.确定递归公式（关系）
2.确定边界(终止)条件

当一个函数调用另一个函数过程时：
在运行被调用函数之前，系统需要完成三件事：
（1）将所有的实在参数，返回地址等信息传给被调用的函数保存好；
（2）为被调用的函数的局部变量分配内存空间；
（3）将控制转移到被调用函数。
被调用函数结束后返回调用之前，系统也要完成三件事：
（1）保存被调用函数的计算结果；
（2）释放被调用函数的内存空间（局部变量失去作用，内存空间被回收）；
（3）依照被调用函数保存的返回地址将控制转移到调用函数。
当有嵌套调用时，要按照“后调用先返回”的原则。
上述过程之间的信息传递和控制转移是通过系统栈来完成的。

系统栈保存了函数的返回地址（上一层中，调用结束后下一步运行的位置）和局部变量。

递归过程时，系统有一个“递归工作栈”：主过程为第 0 层，第一次调用进入第 1 层，从第 i 层调用该过程进入到下一层，即第 i+1 层。反之，退出第 i 层应回到上一层，即第 i-1 层。

二.带有返回值的递归函数

递归函数返回需要的某个值，如:int。

【例 1】猴子吃桃问题

小猴摘了很多桃子。
第一天吃了一半又多吃一个；
第二天又吃掉一半再多吃一个；
……
每天都吃掉前一天桃子数量的一半再多吃一个。
如此下去，到第十天恰好还剩一个桃子。
问第一天小猴摘了多少桃子？
【分析】
f(i)为第 i 天剩下的桃子数量，f(i+1)为第 i+1 天剩下桃子的数量，则有关系：
f(i+1)=f(i)/2+1
得到递推关系式：f(i)=2(f(i+1)+1)
边界条件为：i=10 时，f(10)=1.
求 f(1)=?.
使用递归函数实现如下：

```
//猴子吃桃问题
```

【例 2】1755:菲波那契数列

（Fibonacci sequence）
描述
菲波那契数列是指这样的数列：数列的第一个和第二个数都为 1，接下来每个数都等于前面 2 个数之和。
给出一个正整数 a，要求菲波那契数列中第 a 个数是多少。
输入
第 1 行是测试数据的组数 n，后面跟着 n 行输入。每组测试数据占 1 行，包括一个正整数 a(1 <= a <= 20)
输出
输出有 n 行，每行输出对应一个输入。输出应是一个正整数，为菲波那契数列中第 a 个数的大小
样例输入

```
4
5
2
19
```

1

样例输出

5

1

4181

1

【分析】

递推关系：fib(i)=fib(i-1)+fib(i-2);

边界条件：fib1)=1,fib(2)=1.

思考为什么需要两个边界条件？

【例 3】3089:爬楼梯

描述

树老师爬楼梯，他可以每次走 1 级或者 2 级，输入楼梯的级数，求不同的走法数。

例如：楼梯一共有 3 级，他可以每次都走一级，或者第一次走一级，第二次走两级，也可以第一次走两级，第二次走一级，一共 3 种方法。

输入

输入包含若干行，每行包含一个正整数 N，代表楼梯级数，1 <= N <= 30

输出

不同的走法数，每一行输入对应一行输出

样例输入

5

8

10

样例输出

8

34

89

【分析】

递推关系：f(i)=f(i-1)+f(i-2);

边界条件：f(1)=1,f(2)=2.

例 3 和例 4 递推关系一样，边界条件不一样，结果也不一样，所以边界条件非常关键。

【例 4】1788:Pell 数列

描述

Pell 数列 a1, a2, a3, ...的定义是这样的，a1 = 1, a2 = 2, ... , $a_n = 2 * a_{n-1} + a_{n-2}$ (n > 2)。

给出一个正整数 k，要求 Pell 数列的第 k 项模上 32767 是多少。

输入

第 1 行是测试数据的组数 n，后面跟着 n 行输入。每组测试数据占 1 行，包括一个正整数 k (1 ≤ k < 1000000)。

输出

n 行，每行输出对应一个输入。输出应是一个非负整数。

样例输入

2

1

8

样例输出

1

408

【例 5】666:放苹果

描述

把 M 个同样的苹果放在 N 个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的分法？（用 K 表示）5，1，1 和 1，5，1 是同一种分法。

输入

第一行是测试数据的数目 t（0 <= t <= 20）。以下每行均包含二个整数 M 和 N，以空格分开。1<=M，N<=10。

输出

对输入的每组数据 M 和 N，用一行输出相应的 K。

样例输入

1

7 3

样例输出

8

【分析】

定义：f(i,j)位 i 个苹果放到 j 个盘子的放法，分为两种方案：

方案 1：保证每个盘子至少有 1 个苹果的方案：先拿出 j 个苹果，每个盘子放一个，保证每个盘子里有一个苹果，然后把剩下的 i-j 个苹果再随意放到 j 个盘子了里，方案数是 f(i-j,j);

方案 2：保证至少有一个盘子是空的：任选其中一个盘子一个都不放（空盘子），然后把 i 个苹果放到其余的 j-1 个盘子了，方案是 f(i,j-1).

按分类的加法原理，所以总的方案是：f(i,j)=f(i-j,j)+f(i,j-1)

关键的边界条件：

f(i,j)=0 (i<0)

f(i,j)=1(j=1)

注意：i=0，j=1 时 f(i,j)=1;

尝试不用递归改为递推实现。

【例 6】7592:最大公约数

输入 a 和 b，输出 a 和 b 的最大公约数。

如：

输入：100 75

输出：25

欧几里德算法（又称辗转相除法）

用于计算两个正整数 a，b 的最大公约数。

一般把 a 和 b 的最大公约数记为 gcd(a,b)。

公式：

gcd(a,b)=gcd(b,a mod b)

gcd(a,0)=a

如：gcd(100,75)=gcd(75,25)=gcd(25,0)=25；

代码 1：

//gcd

gcd 函数使用三目运算简写如下：

```
int gcd(int a,int b){ return b==0?a:gcd(b,a%b);}
```

猴子吃桃也可以使用三目运算简写：

三.无返回值的递归函数

函数的作用是完成某个功能:void 类型，往往成为过程，无需返回具体值。

【例 1】利用递归实现输出 1 到 10.

```
#include<iostream>
#include<cstdio>
using namespace std;
void dfs(int i){
    if(i>10) return;
    cout<<i<<" ";
    dfs(i+1);
}
int main(){
    int n;
    dfs(1);
    return 0;
}
```

（1）如何实现输出 10 到 1？

```
#include<iostream>
#include<cstdio>
using namespace std;
void dfs(int i){
    if(i>10) return;
    dfs(i+1);
    cout<<i<<" ";
}
```

```
int main(){
    int n;
    dfs(1);
    return 0;
}
```

（2）阅读下列程序的输出结果：

```
#include<iostream>
#include<cstdio>
using namespace std;
void dfs(int i){
    if(i>10) return;
    cout<<i<<" ";
    dfs(i+1);
    cout<<i<<" ";
}
int main(){
    int n;
    dfs(1);
    return 0;
}
```

（3）阅读下列程序的输出结果：

```
#include<iostream>
#include<cstdio>
using namespace std;
void dfs(int i){
    if(i>4) return;
    dfs(i+1);
    cout<<i<<" ";
    dfs(i+1);
}
int main(){
    int n;
    dfs(1);
    return 0;
}
```

（4）阅读下列程序的输出结果：

```
#include<iostream>
#include<cstdio>
using namespace std;
void dfs(int i){
    if(i<=0) return;
    dfs(i-1);
    cout<<i<<" ";
    dfs(i-1);
```

```
}
int main(){
    int n;
    dfs(3);
    return 0;
}
```

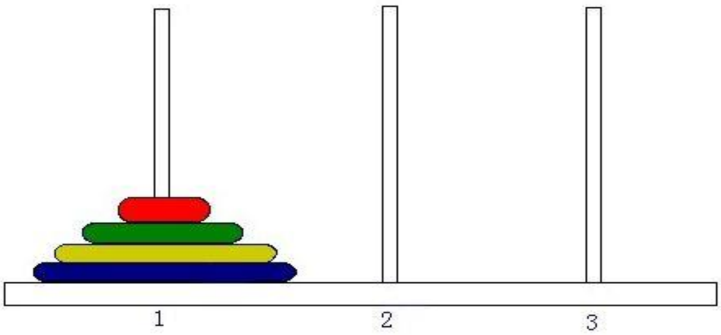
【例 2】输入 n，输出 n 的二进制

输入 n，输出 n 的二进制表示。
输入：10
输出：1010

//n 转换为二进制

【例 3】Hanoi（汉诺塔）问题

http://www.4399.com/flash/109504_1.htm
问题的提出：
Hanoi 塔由 n 个大小不同的圆盘和 3 根木柱 1,2,3 组成。开始时，这 n 个圆盘由大到小依次套在 1 柱上，如图所示。



现在要求用最少的移动次数把 1 柱上 n 个圆盘按下述规则移到 3 柱上：
(1) 一次只能移一个圆盘；
(2) 圆盘只能在 3 个柱上存放；
(3) 在移动过程中，不允许大盘压小盘。
请编程描述移动的过程。　。
输入：
3
输出：
1 : 1-->3
2 : 1-->2
1 : 3-->2
3 : 1-->3
1 : 2-->1
2 : 2-->3

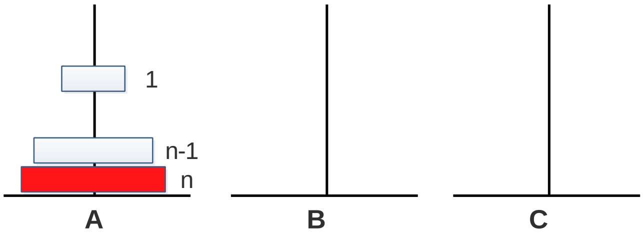
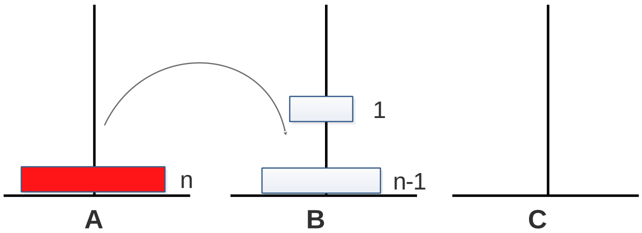
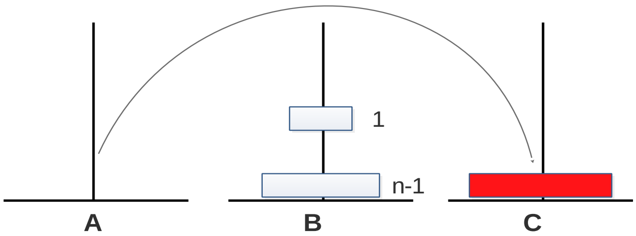
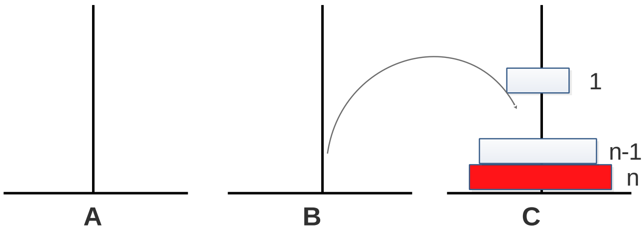
```
1 : 1-->3
```

【算法分析】：

n 个盘子的情况：

用过程 dfs(n,a,b,c)表示把 n 个盘子从 a 经过 b 移动到 c。a,b,c 是有顺序的变量。

调用时： dfs(n,1,2,3)

初始状态 dfs(n,1,2,3)	
第一步：把 A 上面的 n-1 个盘子按照移动规则移动到 B 上。 dfs(n-1,a,c,b)	
第二步：把 A 上最下面的第 n 个盘子从 A 移动到 C。 n:a→c	
第三步：把 B 上面的 n-1 个盘子按照移动规则移动到 C 上。 dfs(n-1,b,a,c)	

参考代码：

```
//汉诺塔
#include<cstdio>
#include<iostream>
using namespace std;
void dfs(int i,int a,int b,int c){
    if(i==1)
        cout<<i<<":"<<a<<"->"<<c<<endl;
    else{
        dfs(i-1,a,c,b);
        cout<<i<<":"<<a<<"->"<<c<<endl;
        dfs(i-1,b,a,c);
```

```
    }
}
int main(){
    int n;
    cin>>n;
    dfs(n,1,2,3);
    return 0;
}
```

要模拟过程

或者：

```
#include<cstdio>
#include<iostream>
using namespace std;
void dfs(int i,int a,int b,int c){
    if(i==0) return;
    dfs(i-1,a,c,b);
    cout<<i<<":"<<a<<"->"<<c<<endl;
    dfs(i-1,b,a,c);
}
int main(){
    int n;
    cin>>n;
    dfs(n,1,2,3);
    return 0;
}
```

【例 4】6261:汉诺塔问题

```
//汉诺塔:柱子编号由 1,2,3 变为了 a,b,c。
#include<cstdio>
#include<iostream>
using namespace std;
string a,b,c;
void dfs(int i,string a,string b,string c){
    if(i==0) return;
    dfs(i-1,a,b,c);
    cout<<a<<"->"<<i<<"->"<<b<<endl;
    dfs(i-1,c,a,b);
}
int main(){
    int n;
    cin>>n>>a>>b>>c;
    dfs(n,a,b,c);
    return 0;
}
```

思考：只输出最少移动次数是多少？

【例 5】自然的分解和

描述

给出一个正整数 a，要求分解成若干个正整数的乘积，即 a = a1 + a2 + a3+ ... + an，并且 1 <= a1 <= a2 <= a3 <= ... <= an，问这样的分解的种数有多少。注意到 a = a 也是一种分解。

输入

n(n<=20)

输出

分解方案

样例输入

7

样例输出

```
1:7
2:1+6
3:1+1+5
4:1+1+1+4
5:1+1+1+1+3
6:1+1+1+1+1+2
7:1+1+1+1+1+1+1
8:1+1+1+2+2
9:1+1+2+3
10:1+2+4
11:1+2+2+2
12:1+3+3
13:2+5
14:2+2+3
15:3+4
```

oj-1751:分解因数

描述

给出一个正整数 a，要求分解成若干个正整数的乘积，即 a = a1 * a2 * a3 * ... * an，并且 1 < a1 <= a2 <= a3 <= ... <= an，问这样的分解的种数有多少。注意到 a = a 也是一种分解。

输入

第 1 行是测试数据的组数 n，后面跟着 n 行输入。每组测试数据占 1 行，包括一个正整数 a (1 < a < 32768)

输出

n 行，每行输出对应一个输入。输出应是一个正整数，指明满足要求的分解的种数

样例输入

2

2

20

样例输出

1

4

分析：

当前状态：n=a[1]*..*a[dep]，下一步继续分解 a[dep]=a[dep]*a[dep+1]为两项的乘积。

从 a[1]=n 开始。预设 a[0]=2。

输出分解方案：

```
1
200
1:200
2:2*100
3:2*2*50
4:2*2*2*25
5:2*2*2*5*5
6:2*2*5*10
7:2*4*25
8:2*4*5*5
9:2*5*20
10:2*10*10
11:4*50
12:4*5*10
13:5*40
14:5*5*8
15:8*25
16:10*20
16
```

【例 6】数的计算(noip2001 普及组)

【问题描述】

对于给定的一个自然数 n(n<=1000)，要求找出所有具有下列性质的数(包含输入的自然数 n)。

把自然数 n 按照如下方法进行处理：

1.不作任何处理;

2. 在它的左边加上一个自然数,但该自然数不能超过原数的一半;

3. 加上数后,继续按此规则进行处理,直到不能再加自然数为止.

如 n= 6

满足条件的数为共有 6 个：

6 ，16，26，126，36，136

【样例输入】

6

【样例输出】

1:6

2:16

3:26

4:126

5:36

6:136

【例 7】8758:2 的幂次方表示

描述

任何一个正整数都可以用 2 的幂次方表示。例如：

137=27+23+20

同时约定方次用括号来表示，即 ab 可表示为 a(b)。由此可知，137 可表示为：

2(7)+2(3)+2(0)

进一步：7=22+2+20（21 用 2 表示）

3=2+20

所以最后 137 可表示为：

2(2(2)+2+2(0))+2(2+2(0))+2(0)

又如：

1315=210+28+25+2+1

所以 1315 最后可表示为：

2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)

输入

一个正整数 n（n≤20000）。

输出

一行，符合约定的 n 的 0，2 表示（在表示中不能有空格）。

样例输入

137

样例输出

2(2(2)+2+2(0))+2(2+2(0))+2(0)

来源 NOIP1998 复赛 普及组 第一题

注意递归的分层次结构：从初始构造搜索树的形式，好理解。