第一题

无穷数列1,1,2,3,5,8,13,21……称为Fibonacci数列。它的递归形式定义为：  
当n=1,2时，F(n)=1；当n＞2时，F(n)=F(n-1)+F(n-2)；编写一个递归函数计算Fibonacci数列的第n项。  
本题包含多组测试数据。  
输入格式说明：每组测试数据输入占一行，包括一个正整数n，n在1~40之间（闭区间），遇文件尾测试结束。  
输出格式说明：每组测试样例的输出结果占一行，输出F(n)的值。

格式注意：输入的要求也是循环输入，因而需要while(scanf(“%d”,&n)!=EOF)来循环输入数字n来得到第n项的结果，这题的n范围在1~40，因此结果是int型就足够了，注意一次输入一次输出，注意换行问题。这题需要用递归来解决，务必弄清递归的出口，并理解每一项之间的关系，从而将关系转化为代码。

测试输入输出：

第一次输入：28

第一次输出：317811

第二次输入：15

第二次输出：610

第二题

定义宏Swap(x,y)，用于交换两个参数x和y的值，并编写测试程序。  
本题包含多组测试数据,组数小于10000组。  
输入格式说明：每组测试数据输入占一行，包括两个整数，依次为x，y，整数范围皆在-10000~10000之间（闭区间），遇文件尾测试结束。  
输出格式说明：对于每组测试数据，输出交换前x、y的值和交换后x、y的值。需要在第i组测试数据结果输出前添加一行提示信息:“CASE i:”，表示是第i组测试数据对应的结果，相邻两组测试样例的输出结果之间空一行。

格式注意：这题要求是用宏定义swap函数，因而记住在#include <stdio.h>下面#define Swap(x,y) …来定义你的swap函数，“遇文件尾结束”则是每次要你输入两个数字,while(“%d%d”,&a,&b)!=EOF,每次输出都要一个case number;所以你要定义number,每次循环number+=1;注意每次循环一次后都会空一排，所以每次循环输出的结构基本是printf(“before…\n”)和printf(“After…\n”)加上printf(“\n”)这样。注意单词大小写以及数字之间的空格。

测试输入输出：

第一次输入：-9959 8467

第一次输出：Case 1:

Before Swap:a=-9959 b=8467

After Swap:a=8467 b=-9959（注意它和下次输出有个空排）

第n次输入：-3666 -3501

第n次输出：Case n:

Before Swap:a=-3666 b=-3501

After Swap:a=-3501 b=-3666

第三题

输入正整数n和k，输出n中从右端开始的第k个数字的值（k从1开始）。将求n中右端第k个数字定义成函数，如果k超过了n的位数，则函数返回-1；否则返回n中第k个数字。

本题包含多组测试数据(由于平台的关系，建议不要采用pow()库函数)。

输入格式说明：每组测试数据输入占一行，包括两个正整数，依次为n，k，

n在1~4294967293之间（闭区间），遇文件尾测试结束。

输出格式说明：每组测试样例的输出结果占一行，输出函数返回结果。

格式注意：这题的输出格式比较简单，输入区间为整数，可以将n定义为无符号数，这里偷偷告诉你们后台测试数据n最大不过26963，因此unsigned int n即可。遇文件尾结束，循环输入不再赘述。这里尽量不要用数组来装n的各位，数组的值未初始化时默认值是0，所以你找超出位数的对应位返回的不是-1而是0，建议先确定n的位数再对它操作。

测试输入输出：

第一组输入：26963 9

第一组输出：-1

第二组输入：321 3

第二组输出：3

第四题

n(0)是一个给定的正整数，对于i=0,1,2,...,定义：（1）若n(i)是偶数，则n(i+1)=n(i）/2;（2）若n(i)是奇数，则n(i+1)=3n(i)+1;(3)若n(i)是1，则序列结束。

本题包含多组测试数据。

输入格式说明：每组测试数据输入占一行，包括一个正整数n0，n0在1~4000000000之间（闭区间），遇文件尾测试结束。

输出格式说明：输出冰雹数序列(用函数实现)以及序列数的总个数，每组测试样例的输出结果后空一行.

格式注意：这题后台测试的数据达到千万级别，建议用long long来定义各个数，输入一次按下回车得到相应的结果后与下一个结果之间有一个空排。这题注意不要用递归来做，这题测试数字太大而递归的速度太慢，系统可能跑不出来会报错。

测试输入输出：

第一组输入：18468

第一组输出：

Hailstone generated by 18468:

18468 9234 4617 13852 6926 3463 10390 5195 15586 7793 23380 11690 5845 17536 8768 4384 2192 1096 548 274 137 412 206 103 310 155 466 233 700 350 175 526 263 790 395 1186 593 1780 890 445 1336 668 334 167 502 251 754 377 1132 566 283 850 425 1276 638 319 958 479 1438 719 2158 1079 3238 1619 4858 2429 7288 3644 1822 911 2734 1367 4102 2051 6154 3077 9232 4616 2308 1154 577 1732 866 433 1300 650 325 976 488 244 122 61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1

Number of hailstone generated:111（这里不仅有个换行，与下个输出还有个空排）

第二组输入：6335

第二组输出：Hailstone generated by 6335:

6335 19006 9503 28510 14255 42766 21383 64150 32075 96226 48113 144340 72170 36085 108256 54128 27064 13532 6766 3383 10150 5075 15226 7613 22840 11420 5710 2855 8566 4283 12850 6425 19276 9638 4819 14458 7229 21688 10844 5422 2711 8134 4067 12202 6101 18304 9152 4576 2288 1144 572 286 143 430 215 646 323 970 485 1456 728 364 182 91 274 137 412 206 103 310 155 466 233 700 350 175 526 263 790 395 1186 593 1780 890 445 1336 668 334 167 502 251 754 377 1132 566 283 850 425 1276 638 319 958 479 1438 719 2158 1079 3238 1619 4858 2429 7288 3644 1822 911 2734 1367 4102 2051 6154 3077 9232 4616 2308 1154 577 1732 866 433 1300 650 325 976 488 244 122 61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1

Number of hailstone generated:156

第五题

编写一个模拟“投掷双骰子”的游戏程序。游戏规则：每轮投两次，取两次的和，第一轮若和为7或11则获胜，游戏结束；若和为2,3，或12则输了，失败结束；若和为其他数字，则将此值作为自己的点数，继续第二轮，第三轮……直到某轮的和等于该点数则获胜，若出现和为7，则输掉游戏。模拟每掷一次骰子的随机取数规则：输入一个非负整数，称为启动数，你需要计算得到启动数各位数之和记为sum，现将1~6这6张数字卡片按顺时钟方向摆放一个环，一只蚂蚁每次从数字为1的卡片摆放的位置出发，这时你告诉蚂蚁按顺时钟方向走sum步，蚂蚁最终停留位置上的卡片数字即视为本次掷骰子得到的点数。

本题包含多组测试数据。

输入格式说明：第一行输入n表示测试样例的个数，后面紧接着有n行输入，每行输入包含两个非负整数记为a，b，整数范围在0~10000之间（闭区间），视为第一轮两次投掷的启动数，若进行到了第n轮时，则第n轮两次投掷的启动数分别看作是a+n-1,b+n-1；

输出格式说明：每组测试样例的输出结果占一行，输出本次游戏的最终结果。

注意：测试数据有100多组

格式注意：这题测试数据最大不过一万，因而int型即可。注意字母大小写和感叹号的中英文，换行等问题即可。每个输出之间没有空排，只需换行。第一次输入你要测试的数据组数然后进行while(scanf()!=EOF){}的输入输出循环操作就行了。

测试输入输出：（测2组数据，输入2）

第一组输入：1234 1

第一组输出：success!

第二组输入：1 42

第二组输出：fail!