**Beautiful String**

**Time Limit : 3000/1000ms (Java/Other)   Memory Limit : 65535/32768K (Java/Other)**

**Problem Description**

ACM竞赛围绕字符串的题目数不胜数，这不又来一个字符串的题目需要你去解决。已知：  
第0个字符串：U  
第1个字符串：DU  
第2个字符串：UDDU  
第3个字符串：DUUDUDDU  
第4个字符串：UDDUDUUDDUUDUDDU  
......  
相信你已经发现规律了，没错！就是第i个字符串 = 第i-1个字符串的取反 + 第i-1个字符串;取反(U->D, D->U);  
现在告诉你n和k，让你求得第n个字符串的第k个字符是多少。(注意字符串编号从0开始);

**Input**

多组输入，每组输入两个数字n, k;(0 <= n <= 51, 0 <= k < 2^n);

**Output**

输出up或者down表示所求字符是 U或者D;

**Sample Input**

3 6

51 123456789012345

**Sample Output**

down

up

http://www.cnblogs.com/cxhscst2/p/6522154.html

分类讨论当前所求字符在字符串的前一半还是后一半，算出要改变多少次cnt。  
那么最后看cnt的奇偶性即可。

#include <bits/stdc++.h>

using namespace std;

long long b[60];

int n;

long long l;

int main()

{

b[0] = 1;

for (int i=1;i<=56;++i)

b[i]=b[i-1]\*2;

while (cin>>n>>l)

{

++l;

int cnt=0;

for(;n;--n) //b[n-1]就是当前这第n个数有几位，比如现在是8位数，而b[n-1]就是4，如果l<b[n-1]那么就是前半段的，就是反转过

if(l>b[n-1])

l-=b[n-1];

else ++cnt;//这是在算他翻转了几次

puts(cnt%2 ? "down":"up");//变了奇数次是down

}

return 0;

}