题意：

你和一个人玩游戏，给你一个数字n，每次操作可以从2~9中任选一个数字，并把它与p相乘，（游戏开始时p=1）

两人轮流操作，当一个人操作完后p>=n，这个人就是胜者。

**思路：1)找规律**

**如果n是 2 ~ 9 ，Stan 必胜。**

**如果输入是 10~18 ，不管第一次Stan 乘的是什么，Stan肯定在 2 ~ 9 之间，**

**无论stan乘以什么，Ollie乘以大于1的数都都能超过 10 ~ 18 中的任何一个数。Ollie 必胜。**

**如果输入是 19 ~ 162，那么这个范围是 Stan 的必胜态。**

**如果输入是 163 ~ 324 ，这是又是Ollie的必胜态。**

**............**

**必胜态是有规律可循的。**

**如果＂我方＂首先给出了一个在n不断除１８后的得到不足１８的**

**数m，＂我方＂就可以取得胜利，然而双方都很聪明，所以这样胜负就决定于n了，**

**如果n不断除１８后的得到不足１８的数m，**

**若１＜m＜＝９则先手胜利，**

**若９＜m＜＝１８则后手胜利．**

解题思路：

由于每次都是从p=1开始的，所以只要判断每个游戏中1为必败点还是必胜点即可。（以下各式 / 均为取上整）

依照上面所提到的算法，将终结位置，即[n,无穷]标记为必败点；

然后将所有一步能到达此必败段的点标记为必胜点，即[n/9,n-1]为必胜点；

然后将只能到达必胜点的点标记为必败点，即[n/9/2,n/9-1]为必败点；

重复上面2个步骤，直至可以确定1是必胜点还是必败点。

下面正式分析一下，2--9内。s必胜；超过9，那么s必定选2，也就是说在10--9\*2内，o必胜；超过18，s就可以通过控制第一次的取值来获胜，最大就是9\*2\*9；我们可以看到，每乘18就会循环一轮，所以只需要让n不断除以18，最后判断是落在1--9还是10--18里就可以了，注意，由于存在除不尽的情况，需要将n设置为double

#include<stdio.h>

#include<iostream>

#include<cstring>

#include<set>

#include<map>

#include<cmath>

typedef long long ll;

using namespace std;

int main()

{

//freopen("input.txt","r",stdin);

int n;

while(cin>>n)

{

int x(0);

while(n>1)

{

if(x&1)

n=ceil(n\*1.0/2);

else n=ceil(n\*1.0/9);

x++;

}

if(x&1)

puts("Stan wins.");

else puts("Ollie wins.");

}

return 0;

}