这是第五次实践报告

在实验5.2与实验5.3中，首先都应该用eval（）函数录入数据，实验5.2中，python没有指定奇数偶数的保留函数，故应该定义一个函数用来判断奇偶性。而判断整数，浮点，复数甚至还是字符，都有专门的保留函数可用。

一个好用的语句如下：

isinstance(a，保留字)

当数据a是保留字指定的类型时，返回Ture

isinstance() 函数来判断一个对象是否是一个已知的类型，类似 type()

整数int 浮点float 复数complex

1. 也可以用type判断  
   def isNum(n):
2. try:
3. n=type(eval&*#40;n&#41;)*
4. if n==type(1):
5. return True
6. elif n==type(1.0):
7. return True
8. elif n==type(1+1j):
9. return True
10. else:
11. return False
12. except:
13. return False
14. print(isNum(input("输入：")))

实现multi()函数,适当修改教材p129实例即可，关键是修改‘+=’为‘\*=’符

斐波拉契函数的简单写法

1. def fib(n):
2. if n==1:
3. return 1
4. if n==2:
5. return 1
6. return fib(n-1)+fib(n-2)

关键是建立分支和其对应的嵌套。

汉诺塔的难点是要在于要理解一次只能移动一个盘子，而且必须是下面最大的那张。假设用N个盘子在A柱，

则

移动N-1张至B柱，剩下最后最大的移到C柱。此时还有N-1张在B柱上（A柱此时是空的）

将N-2张至A柱，B柱剩下的一张移到C柱，这样就实现了一次递归。

接下来就是A与B柱上的碟子反复调换，越来越少，C柱上一张张地增加

关键就是实现这部分的程序，同时要注意move()函数的调用

move函数可以以非常简单的方式将左值引用转换为右值引用。

如下

如果n=1时，即就一层时，就只需要将a1从A移动C就可以了

　　当n=2时， move(n-1, a, c, b)  就将最上层(a1)从A移动到B

　　　　　　　move(1, a, b, c) 将A的最后一个(a2)移动到C

　　　　　　　move(n-1, b, a, c) 将之前移动到B的最上层(a1)移动到C

　　　　　　　完成

关于科赫曲线，仅需要修改实例代码8.1中的第七句，将[0,60,-120,60]:

改为[0,-60,120,-60]

即可以实现向右下画