

求n的阶乘

```
def factorial1(n):
    if n < 2:
        return 1
    return factorial1(n-1) * n

def factorial2(n, fac=1):
    fac = fac * n
    if n < 2:
        return fac
    return factorial2(n-1, fac)

n = 10
print(factorial1(n))
print(factorial2(n))
```

将一个数逆序放入列表中，例如1234 => [4,3,2,1]

```
# 递归取字符
data = str(1234)

def revert(x):
    if x == -1:
        return []
    return [data[x]] + revert(x-1)

print(revert(len(data)-1))

# 递归切片
data = str(1234)

def revert(x, target=[]):
    if x:
        target.append(x[-1])
        revert(x[:-1])
    return target

print(revert(data))
```

```
# 使用数字整除取模递归
def revert(x, target=None):
    if target is None:
        target = []
    x, y = divmod(x, 10)
    target.append(y)

    if x == 0:
        return target
    return revert(x, target)

print(revert(123045))
```

解决猴子吃桃问题

猴子第一天摘下若干个桃子，当即吃了一半，还不过瘾，又多吃了一个。第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第10天早上想吃时，只剩下一个桃子了。求第一天共摘多少个桃子

思路：
假设猴子摘了x个桃

```
d1 x // 2 - 1
d2 d1 // 2 - 1
d3 d2 // 2 - 1
...
d9 d8 // 2 - 1
d10 1
```

```
def peach(days=10):
    if days == 1:
        return 1
    return (peach(days-1)+1)*2

print(peach())
```

注意这里必须是10，因为return (peach(days-1)+1)*2立即拿不到结果，必须通过再一次进入函数时判断是不是到了最后一天。也就是当前使用的值是由下一次函数调用得到，所以要执行10次函数调用

换种方式表达

```
def peach(days=1):
    if days == 10:
        return 1
    return (peach(days+1)+1)*2

print(peach())
```

