能被3整除，不能被5整除

此处用的是遍历，也可以用数学关系直接算

In [1]:

```python
def solution():
    cnt = 0
    for i in range(1,1001):
        if i%3==0 and i%5!=0:
            cnt+=i
    print(cnt)
solution()
```

133668

syn序列

直接用while分情况处理就行

In [2]:

```python
def solution(x):
    while x!=1:
        print(x,end = ' ')
        if x%2==0:
            x=x//2
        else:
            x = 3*x+1
    print(1)
solution(5)
```

5 16 8 4 2 1

找零问题--如果问最少多少张的话就是最经典的动态规划

- 零钱设置合理，使用贪婪
- 零钱设置不合理，使用动态规划(因为需要具体的每种多少张，所以相对复杂)

In [3]:

```python
def zhaoling_tanlan(money):
    if money<0 or money>100 or money!=int(money):
        return -1
    money = 100-money
    zhibi_list = [1,2,5,10,20,50]
    for i in range(len(zhibi_list)-1,-1,-1):
        cur = money//zhibi_list[i]
        money = money%zhibi_list[i]
        if cur!=0:
            print('找零',cur,'张',zhibi_list[i],'元纸币')
zhaoling_tanlan(6)
```

找零 1 张 50 元纸币
找零 2 张 20 元纸币
找零 2 张 2 元纸币

In [4]:

```python
# dp算法如果想要得到找零组合，需要存储一个较大的矩阵
zhibi_list = [2,3,5,10,19]
def zhaoling_dp(money,zhibi_list):
    zhibi_list.sort()
    # 维护一个5维的矩阵，第一维记录多少个硬币，后面记录具体每种的数量(这里money+1是为了下标方便)
    res = [[0 for i in range(len(zhibi_list)+1)] for j in range(money+1)]
    for i in range(money+1):
        curmin = 999999999999999
        for j in range(len(zhibi_list)):
            item = zhibi_list[j]
            if i-item<0:
                break
            if i==item:
                res[i] = [0]*(len(zhibi_list)+1)
                res[i][0] = 1
                res[i][j+1] =1
                break
            if res[i-item][0]>0 and res[i-item][0]+1<curmin:
                curmin = res[i-item][0]+1
                res[i] = res[i-item][:]
                res[i][0]+=1
                res[i][j+1]+=1
    for i in range(len(zhibi_list)):
        if sum(res[-1])==0:
            print('无法找零')
            break
        elif res[-1][i+1]!=0:
            print('找零',res[-1][i+1],'张',zhibi_list[i],'元纸币')
        else:
            continue

zhaoling_dp(898,zhibi_list)
```

找零 1 张 5 元纸币
找零 47 张 19 元纸币