1. 利⽤管程解决哲学家问题

问题描述：一张圆桌上坐着5名哲学家，每两个哲学家之间的桌上摆一根筷子，桌子的中间是一碗米饭。哲学 家们倾注毕生的精力用于思考和进餐，哲学家在思考时，并不影响他人。只有当哲学家饥饿时， 才试图拿起左、右两根筷子（一根一根地拿起）。如果筷子已在他人手上，则需等待。饥饿的哲 学家只有同时拿起两根筷子才可以开始进餐，当进餐完毕后，放下筷子继续思考。

1. 定位：系统中有5个哲学家进程，5位哲学家与左右邻居对其中间筷子的访问是互斥关系，且这个问题中只有互斥关系

2、角色：五位哲学家  
3、定义信号量并且赋初值

定义互斥信号量数组 chopstick[5]={1,1,1,1,1}用于实现对5个筷子的互 斥访问。并对哲学家按0到4编号，哲学家i左边的筷子编号为i，右边的筷子编号为(i+1)%5。

设置mutex初值为1，保证各哲学家拿筷子这件事必须互斥的执行。这就保证了即使一个哲学家在拿筷子拿到一半时被阻塞，也不会有别的哲学家会继续尝试拿筷子。这样的话，当前正在吃饭的哲学家放下筷子后，被阻塞的哲学家就可以获得等待的筷子了。

1. 写主函数

class monitor():  
 def \_\_init\_\_(self,ID):  
 self.ID=ID  
 def pickup(self):

P(mutex)  
 P(chopstick[self.ID])  
 P(chopstick[(self.ID+1)%5])

V(mutex)  
 def putup(self):  
 V(chopstick[self.ID])  
 V(chopstick[(self.ID + 1) % 5])  
  
def philosopler(i):  
 思考问题  
 monitor(i).pickup()  
 吃东西  
 monitor(i).putup()  
  
if \_\_name\_\_=="\_\_main\_\_":  
 chopstick = [1, 1, 1, 1, 1]

mutex =1

While(true):  
 philosopler(i)