（一）   图书馆有100个座位，每位进入图书馆的读者要在登记表上登记，退出时要在登记表上注销。要几个程序？有多少个进程？（答：一个程序；为每个读者设一个进程）  
  
（1）       当图书馆中没有座位时，后到的读者在图书馆为等待（阻塞）  
（2）       当图书馆中没有座位时，后到的读者不等待，立即回家。  
  
解（1 )  
设信号量：S=100;  MUTEX=1  
P(S)  
P(MUTEX)  
登记  
V(MUTEX)  
阅读  
P(MUTEX)  
注销  
V(MUTEX)  
V(S)  
  
     
解(2)  
设整型变量 COUNT=100;   
信号量：MUTEX=1;  
P(MUTEX);  
IF (COUNT==0)  
{ V(MUTEX);  
   RETURN;  
}  
COUNT=COUNT-1;  
登记  
V(MUTEX);  
阅读  
P(MUTEX);  
COUNT=COUNT+1;  
V(MUTEX);  
RETURN;  
  
  
  
（二）  有一座东西方向的独木桥；用P,V操作实现：  
（1）       每次只允许一个人过桥；  
（2）       当独木桥上有行人时，同方向的行人可以同时过桥，相反方向的人必须等待。  
（3）       当独木桥上有自东向西的行人时，同方向的行人可以同时过桥，从西向东的方向，只允许一个人单独过桥。（此问题和读者与写者问题相同,东向西的为读者，西向东的为写者）。  
（1）解  
设信号量 MUTEX=1  
P (MUTEX)  
  过桥  
V (MUTEX)  
(2)解  
设信号量： MUTEX=1 (东西方互斥)  
      MD=1    (东向西使用计数变量互斥)  
   MX=1    (西向东使用计数变量互斥)  
设整型变量： CD=0  (东向西的已上桥人数)  
        CX=0  (西向东的已上桥人数)  
  
从东向西：  
 P (MD)  
 IF (CD=0)  
 {P (MUTEX)  }  
 CD=CD+1  
 V (MD)  
 过桥  
 P (MD)  
 CD=CD-1  
 IF (CD=0)  
 {V (MUTEX)  }  
 V (MD)

从西向东：  
P (MX)  
IF (CX=0)  
{P (MUTEX)  }  
CX=CX+1  
V (MX)  
过桥  
P (MX)  
CX=CX-1  
IF (CX=0)  
{V (MUTEX)  }  
V (MX)  
  
(3) 解：从东向西的，和（2）相同；从西向东的和（1）相同。  
  
（三） 有一个俱乐部，有甲乙两个服务员，当顾客有请求时，甲负责送烟，乙负责送火，无顾客请求时，服务员睡眠。顾客自己不能带烟和火，当顾客要抽烟时，可请求服务员送烟和火，烟和火还未送到时，顾客必须等待。  
设信号量：SY, SH,CY,CH:初值都为0  
  
甲服务员  
REPEAT  
P(SY)  
送烟  
V(CY)  
UNTIL FALSE  
乙服务员  
REPEAT  
P(SH)  
送火  
V(CH)  
UNTIL FALSE  
顾客  
V(SY)  /\*（请求送烟）\*/  
V(SH)  /\*（请求送火）\*/  
P(CY)  /\* (等烟)  \*/  
P(CH)  /\* (等火)  \*/  
抽烟  
  
（四）一家四人父、母、儿子、女儿围桌而坐；桌上有一个水果盘；  
（1）  当水果盘空时，父亲可以放香蕉或者母亲可以放苹果，但盘中已有水果时，就不能放，父母等待。当盘中有香蕉时，女儿可吃香蕉，否则，女儿等待；当盘中有苹果时，儿子可吃，否则，儿子等待。  
解 设信号量：SE=1 (空盘子)；SA＝0 （放了苹果的盘子）；SB=0 (放了香蕉的盘子)  
  
父亲  
REPEAT  
剥香蕉  
P(SE)  
放香蕉  
V(SB)  
UNTIL FALSE  
母亲  
REPEAT  
削苹果  
P(SE)  
放苹果  
V(SA)  
UNTIL FALSE  
儿子  
P(SA)  
拿苹果  
V(SE)  
吃苹果  
  
女儿  
P(SB)  
拿香蕉  
V(SE)  
吃香蕉  
  
（2） 把（1）改为：儿子要吃苹果时，请母亲放苹果，女儿要吃香蕉时，请父亲放香蕉，（还是盘子为空时才可以放）。  
  
（2）解：再增加两个信号量：SF=0, SM=0  
  
父亲  
REPEAT  
P(SF)  
剥香蕉  
P(SE)  
放香蕉  
V(SB)  
UNTIL FALSE  
母亲  
REPEAT  
P(SM)  
削苹果  
P(SE)  
放苹果  
V(SA)  
UNTIL FALSE  
儿子  
V(SM)  
P(SA)  
拿苹果  
V(SE)  
吃苹果  
  
女儿  
V(SF)  
P(SB)  
拿香蕉  
V(SE)  
吃香蕉  
  
（五）有一个超市，最多可容纳N个人进入购物，当N个顾客满员时，后到的顾客在超市外等待；超市中只有一个收银员。可以把顾客和收银员看作两类进程，两类进程间存在同步关系。写出用P;V操作实现的两类进程的算法（2003年系统设计员考试的题目）  
解：设信号量：S=0，C=0 (顾客与收银员的同步信号量)，M=N  
  
收银员  
P(S)  
收银  
V(C)  
  
顾客  
P(M)  
进入店内购物  
V(S)  
P(C)  
V(M)  
  
（六）有一个理发店，店内共有20个座位供顾客等待理发，（进入理发店的顾客，都在此座位上等待理发，正在理发的顾客不占用此座位），当20个座位坐满了，后到的顾客不等待，立即回家。当没有顾客时，理发师睡眠等待。  
  
解：设信号量：S=0.C=0,MUTEX=1  
设整型变量  SM=20  
  
理发师  
REPEAT  
P(S) -------如无顾客，理发师等待  
V(C)     叫一个顾客理发  
理发  
UNTIL FALSE  
  
顾客  
P(MUTEX)  
IF (SM=0)  
{ V(MUTEX)――――满座，离开，回家  
RETURN  
ELSE  
SM=SM-1―――――空座位数减 1  
V(MUTEX)  
}  
V(S)――――――――通知理发师，增加了一个顾客，如理发师在等待则唤醒他  
P(C) ———————等理发师叫自己理发  
P(MUTEX)  
SM=SM+1―――――被叫到，释放一个空的座位  
V(MUTEX)  
接受理发  
如果此题改为：满座时，顾客等待空座位：则 顾客进程的程序修改如下：  
把SM设为信号量  SM=20  
顾客  
P(SM) ---------------------申请一个座位，无则等待  
V(S)――――――――通知理发师，增加了一个顾客，如理发师在等待则唤醒他  
P(C) ———————等理发师叫自己理发  
V(SM)  
接受理发  
（七）一个盒子，内有黑白两种棋子（数量相等），甲每次从盒子中取出一颗黑子，乙每次从盒子中取出一颗白子，一人取了棋子后，必须等另一方取过棋子方可再取，（可假设甲先取）。  
解：  设信号量：SJ=1,SY=0  
  
甲  
REPEAT  
P(SJ)  
取一颗黑子  
V(SY)  
UNTIL 盒子中无黑子  
乙  
REPEAT  
P(SY)  
取一颗白子  
V(SJ)  
UNTIL 盒子中无白子  
  
  
（八）按要求完成下面的程序。设有三个进程，input进程、compute进程和output进程；它们通过共享一个缓冲区buf的合作关系如下：  
（1）input进程每次输入数据后，把数据送到buf,供compute进程计算和output进程打印；  
（2）comput进程每次从buf取出已输入的可计算的数据进行计算，并当output进程把输入数据打印完成后，把计算结果送入buf供output进程打印；  
（3）output进程每次按顺序把buf中的输入数据和计算结果在打印机上输出。  
解：  
设信号量：sa=1,sb=sc=sd=0, 请把能正确实现这三个进程同步关系的P、V 操作的语句填入下面的程序。

procedure input  
begin  
local data  
repeat  
  get(data);   /\*输入数据到data\*/  
  p(sa);  
  buf=data;  
  (1) V ( sc )                   
  v(sb);  
until false  
end;  
procedure compute  
begin  
locol data  
repeat  
 (2)P( sb )                        data=buf;  
   计算data并把结果保存在data;

(3)  P( sd )                          buf=data;  
   v(sc);  
until false  
end;  
procedure output  
begin  
local data  
repeat  
   P(sc)  
   打印 buf;  
 (4) V( sd )                          p(sc)  
   打印 buf;  
   v(sa);  
until false  
end;