**1. 在生产者—消费者问题中，如果缺少了signal(full)或signal(empty)，对执行结果将会有何影响？**

如果缺少了signal（full），那么生产者进程就不会改变信号量full的值，即使缓冲池中的产品已经放满，信号量full的值依旧是0，这样会导致消费者进程在执行wait（full）是误以为缓冲池为空而自始至终处于阻塞状态，生产者生产完n个产品后阻塞。

如果缺少了signal（empty），那么消费者进程取走产品时empty没有改变，直到缓冲池中所有的产品都被取走，empty的值也一直是0，即使目前缓冲池有空缓冲区，生产者进程在向缓冲池放产品会因申请不到空缓冲区而被阻塞，所以生产者生产完n个产品后阻塞，消费者消费完n个产品后也阻塞。

**2. 在生产者—消费者问题中，如果将两个Wait操作即Wait（full）和Wait（Mutex）互换位置，或者将Signal（Mutex）和Signal（full）互换位置，结果会如何？**

在生产者—消费者问题中，如果将两个wait操作，即wait(full)和wait(mutex)互换位置后，可能引起死锁。考虑系统中缓冲区全满时，若一生产者进程先执行了wait(mutex)操作并获得成功，则当再执行wait(empty)操作时，它将因失败而进入阻塞状态，它期待消费者进程执行signal(empty)来唤醒自己，在此之前，它不可能执行signal(mutex)操作，从而使试图通过执行wait(mutex)操作而进入自己的临界区的其他生产者和所有消费者进程全部进入阻塞状态，这样容易引起系统死锁。

若signal(mutex)和signal(full)互换位置后只是影响进程对临界资源的释放次序，而不会引起系统死锁，因此可以互换位置。

3. 今有三个并发进程R 、M、P,它们共享一个缓冲区，三个进程的执行顺序是R先从输入设备读信息，每读一个记录，把它存放在缓冲区，然后M在缓冲区加工读入的记录，M加工完成最后由P把记录打印输出，这时又可以从头运行R向缓冲区存放下一个记录，如此往复(前趋图如下所示)。试在下述类PASCAL程序中空白处填上信号量初值、Wait、Signal操作实现三个进程正确的并发执行。

c

b

a

解：

var a,b, c :semaphore:=**1，0，0**;

begin

parbegin

R: begin

repeat

Read from i/o

**﹎﹎﹎Wait(a)﹎﹎﹎**;

Add message to buffer;

**﹎﹎﹎Signal(b)﹎﹎﹎**;

until

end

M: begin

repeat

**﹎﹎﹎Wait(b)﹎﹎﹎**;

Get message from buffer;

Process message;

Add message to buffer;

**﹎﹎﹎Signal(c)﹎﹎﹎**;

until

end

P: begin

repeat

**﹎﹎﹎Wait(c)﹎﹎﹎**;

Get message from buffer;

**﹎﹎﹎Signal(a)﹎﹎﹎**;

Print message;

until

end

parend

end

4. 已知有三个并发进程P、Q和R以及一对供存数据的缓冲BufI和BufO，P进程把数据输入BufI，R进程输出BufO中的数据。Q地把BufI中的数据变换后送入BufO，在上述假定之下,使三个进程实现最大并行性。试在下述类PASCAL程序中空缺位置分别填上信号量、信号量初值和Wait、Singal操作实现三个进程正确的并发执行。

BufI

BufO

Program ito;

Var BufI,BufO:buffer;

Var emptyI , fullI , emptyO ,fullO :SEMAPHORE:= 1 , 0 , 1 , 0 ；

begin

parbegin

P:begin

repeat

input from IO;

Wait(emptyI) ；

Add to BufI;

Signal(fullI) ；

until false

end;

Q:begin

repeat

Wait(fullI) ；

Remove from BufI;

Signal(emptyI) ；

transform;

Wait(emptyO) ；

Add to BufO;

Signal(fullO) ；

until false

end;

R:begin

repeat

Wait(fullO) ；

Remove from BufO;

Signal(emptyO) ；

Output ...;

until false

end;

parend

end

5. 现东西两点间有一段单行车道，请设计一个自动管理系统，管理规则如下：当单行车道上有车辆在行驶时，同方向的车可以同时驶入，当另一方向的车必须等待；当单行车道无车辆在行驶时，到达东向端点（西向端点）的车辆可以进入单行车道，但不能从东向端点和西向端点同时驶入；当某方向车辆驶出单行车道且暂无车辆进入单行车道时，应该让另一方向等待的车辆进入单行车道行驶。请用signal(),wait()原语实现改管理系统并保证行驶安全。

解：设互斥访问单行车道的信号量Sab，从西向端点驶入的车辆互斥访问共享变量ba的信号量S2，从东向端点驶入的车辆互斥访问共享变量ab的信号量S1。

Semaphore S1=1,S2=1,Sab=1;

Int ab=ba=0;

Void Pab()

{ while(1)

Wait(S1);

If(ab==0)

Wait(Sab);

ab=ab+1;

signal(S1);

车辆从东向端点驶向西向端点；

Wait（S1）；

ab=ab-1;

if(ab==0)

signal(Sab);

signal(S1);

}

}

Void Pba()

{ while(1)

Wait(S2);

If(ba==0)

Wait(Sab);

ba=ba+1;

signal(S2);

车辆从西向端点驶向东向端点；

Wait（S2）；

ba=ba-1;

if(ba==0)

signal(Sab);

signal(S2);

}

}

Main（）

｛cobegin

｛Pab()；

Pba()；

｝

｝