```
#include<semaphore.h>
#include<stdio.h>
int guard = 0;
 int semaphore value = 0;
 wait ()
{ while (TestAndSet(&guard) == 1);
 if (semaphore value == 0)
 { atomically add process to a queue of processes waiting for the semaphore and set guard
to 0; } else
{ semaphore value--; guard = 0;
}
 }
 signal()
{ while (TestAndSet(&guard) == 1);
 if (semaphore value == 0 && there is a process on the wait queue) wake up the first
process in the queue of waiting processes
else
semaphore value++;
guard = 0;
}
```

```
#include<stdio.h>
#include<semaphore.h>
type resource=monitor var P:
array[0..2]of boolean;
X:condition;
procedure acquire(id: integer, printer-id: integer);
beginif
P[0]and P[1]and P[2]
then X.wait(id)if not P[0]
then printer-id:= 0;
else if not P[1]then printer-id:= 1;
else printer-id:= 2;
P[printer-id] :=true;
endprocedure release(printer-id: integer)beginP[printer-id] :=false;
X.signal;
endbeginP[0] :=P[1] :=P[2] :=false;
end
```