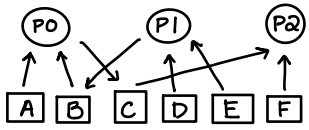
Shanti Upadhyay

Auburn ID: spu0004

1(a).



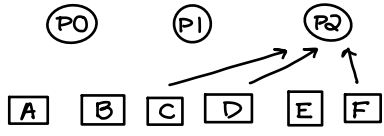
Deadlock occurs because all processes are in a waiting state.

1(b).

```
void P0() {
  while (true) {
    get(B);
    get(C);
    get(A);
  } ...

void P1() {
  while (true) {
    get(D);
    get(E);
    get(B);
  } ...

void P2() {
  while (true) {
    get(F);
    get(D);
    get(C);
  } ...
```



2. Yes the concurrent execution of these two processes can result in one or both being blocked forever. If semWait(R) from the bar() process follows the execution of semWait(S) from the foo() process, both of the semaphore variables would be equal to 0 and would need to wait on the following statements from both bar() and foo() processes.

3. Deadlock avoidance: detecting depends on the programmer. The system itself will attempt to avoid a deadlock if possible.

   Deadlock detection: preventing situations in which the application is hindered. The programmer must implement this function.

   Deadlock prevention: does not need to be implemented, but rather the detection is completed automatically.

4. Deadlocking only occurs if a process is unable to access the maximum amount of resources required. If we are given 3 processes, and each process needs at most 2 resources, one of those processes has already reached the maximum.