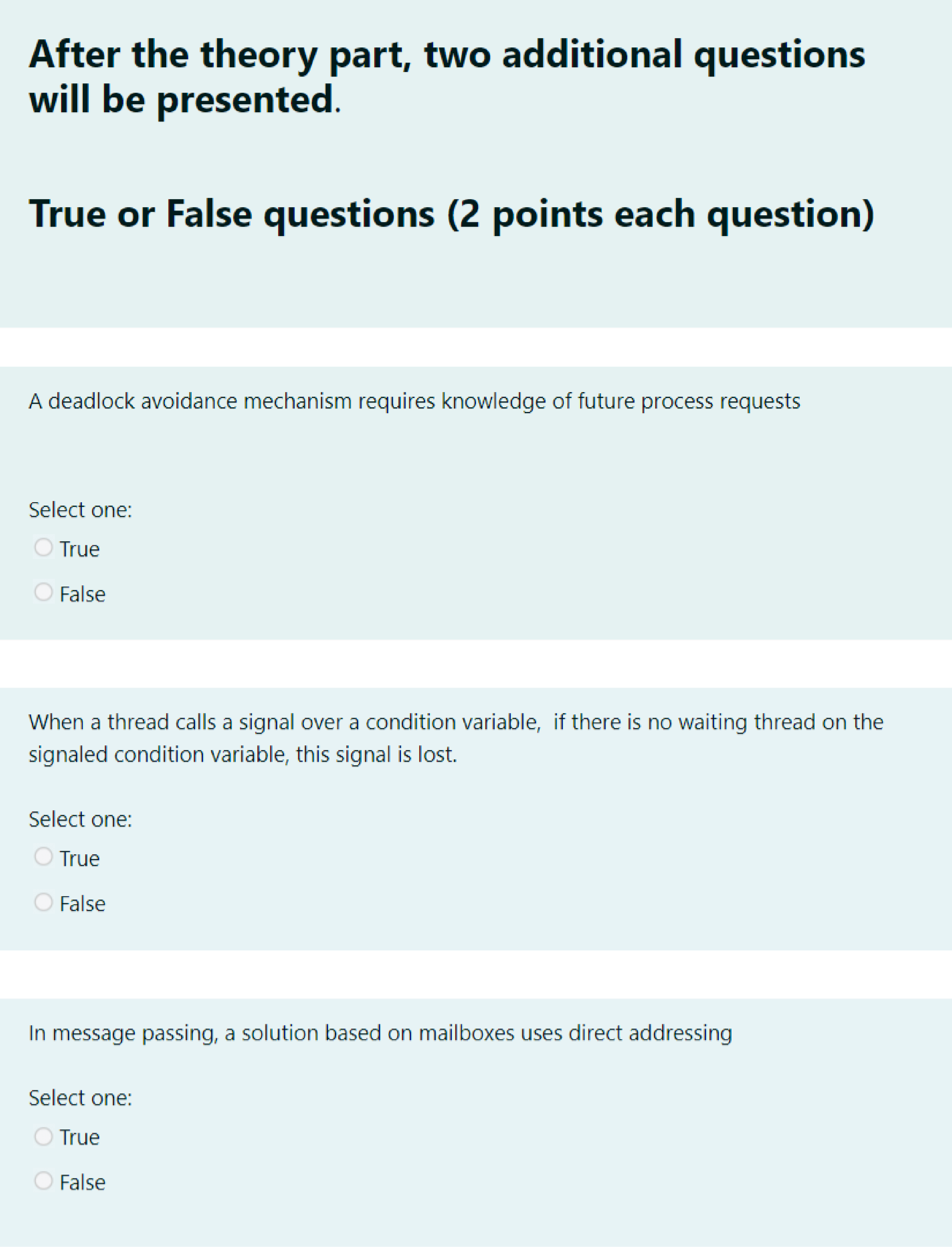


After the theory part, two additional questions will be presented.

True or False questions (2 points each question)

A deadlock avoidance mechanism requires knowledge of future process requests

Select one:

○ True

○ False

When a thread calls a signal over a condition variable, if there is no waiting thread on the signaled condition variable, this signal is lost.

Select one:

○ True

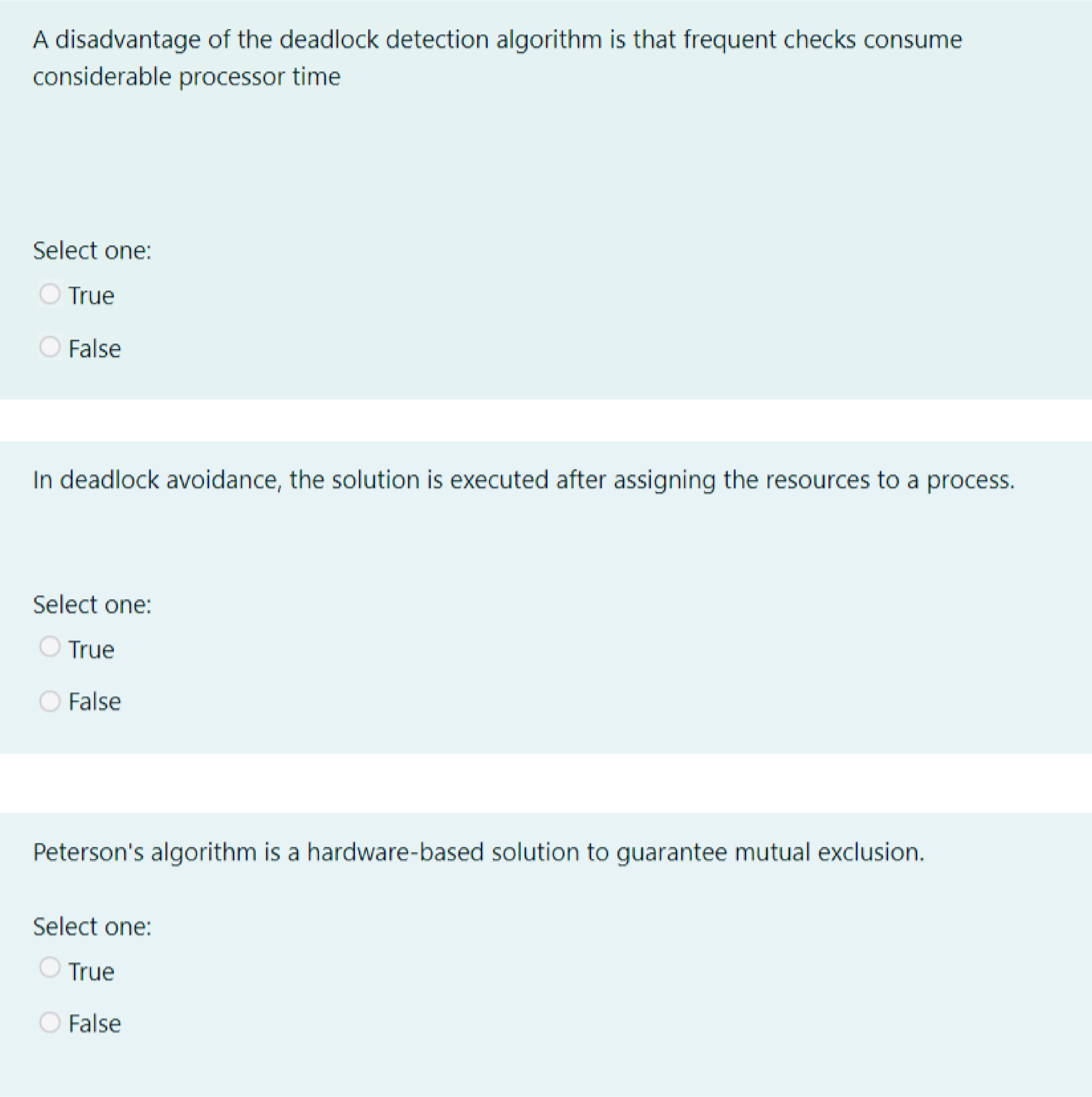
○ False

In message passing, a solution based on mailboxes uses direct addressing

Select one:

○ True

○ False



A disadvantage of the deadlock detection algorithm is that frequent checks consume considerable processor time

Select one:

○ True

○ False

In deadlock avoidance, the solution is executed after assigning the resources to a process.

Select one:

○ True

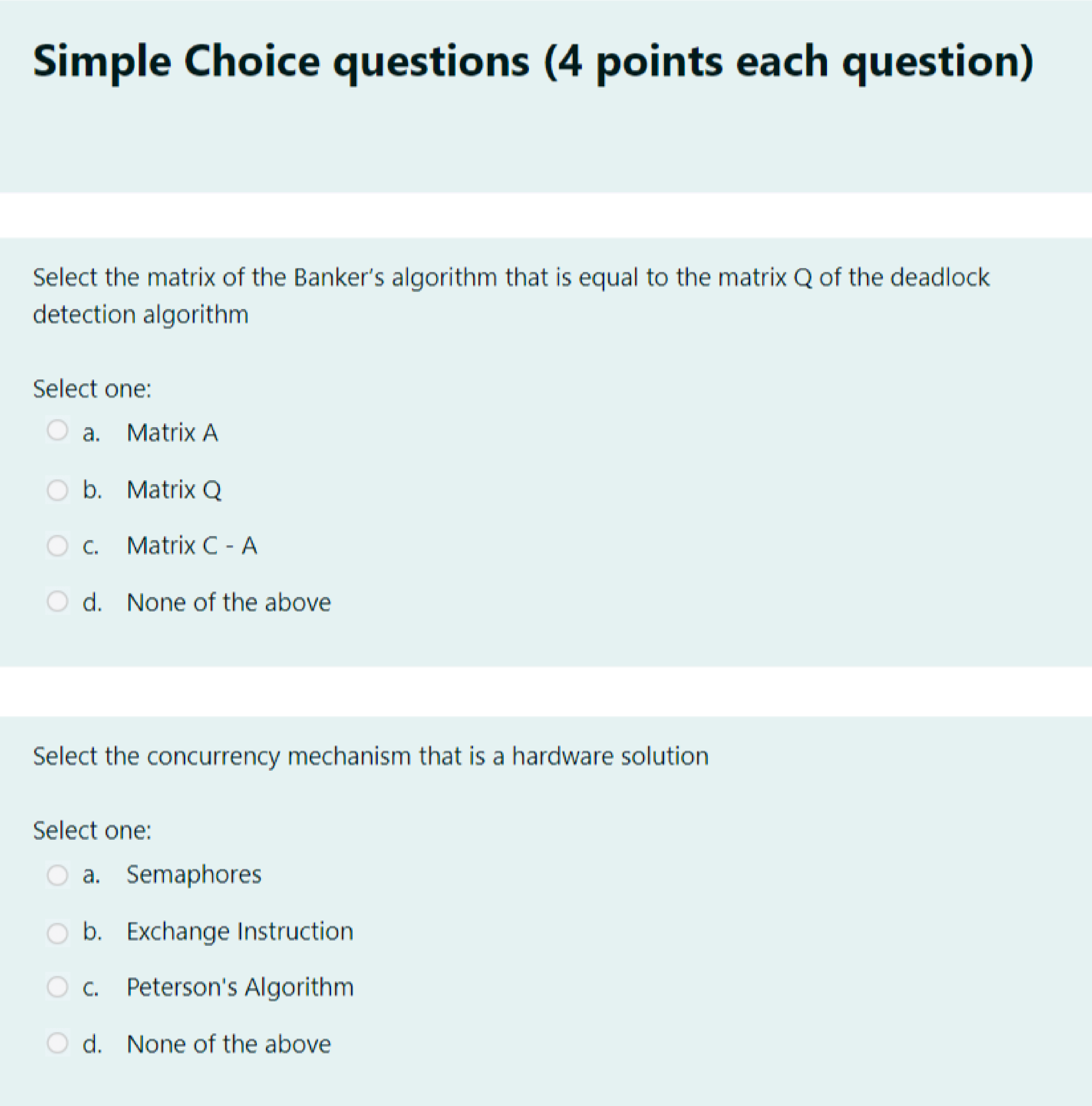
○ False

Peterson's algorithm is a hardware-based solution to guarantee mutual exclusion.

Select one:

○ True

○ False



Simple Choice questions (4 points each question)

Select the matrix of the Banker's algorithm that is equal to the matrix Q of the deadlock detection algorithm

Select one:

○ a. Matrix A

○ b. Matrix Q

○ c. Matrix C - A

○ d. None of the above

Select the concurrency mechanism that is a hardware solution

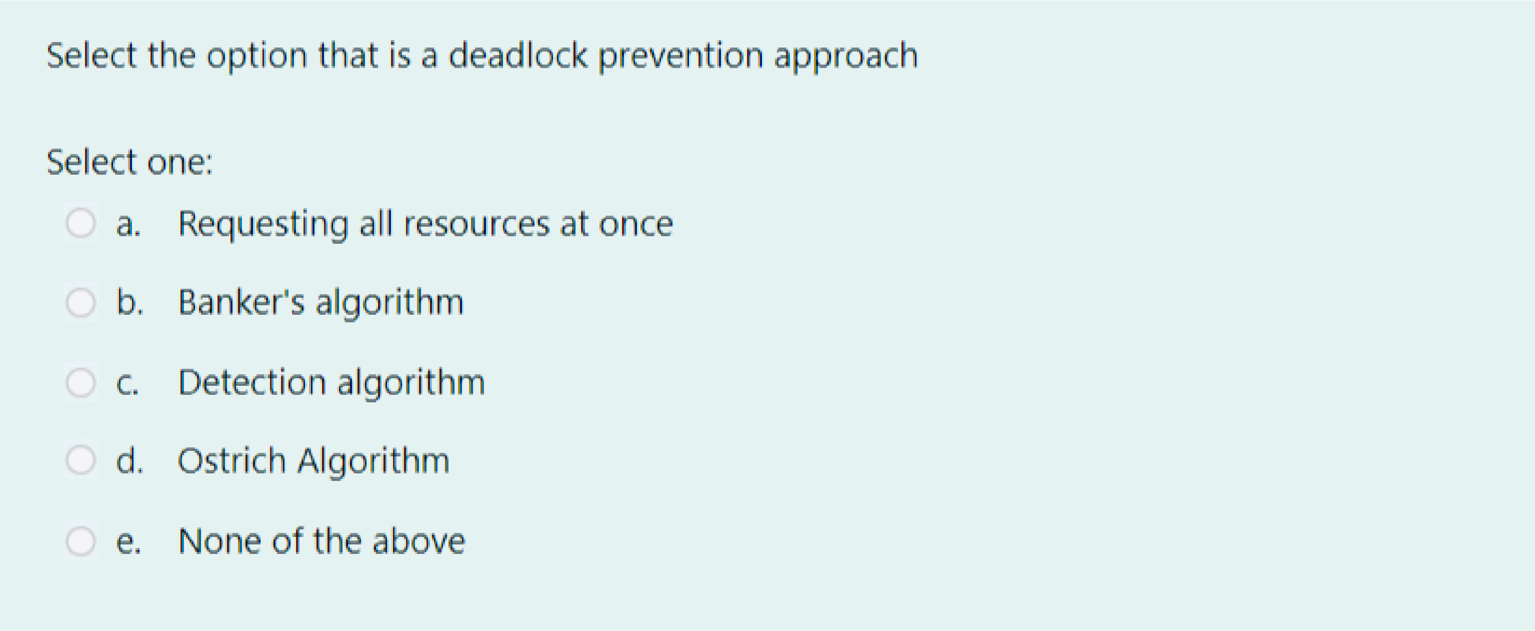
Select one:

○ a. Semaphores

○ b. Exchange Instruction

○ c. Peterson's Algorithm

○ d. None of the above



Select the option that is a deadlock prevention approach

Select one:

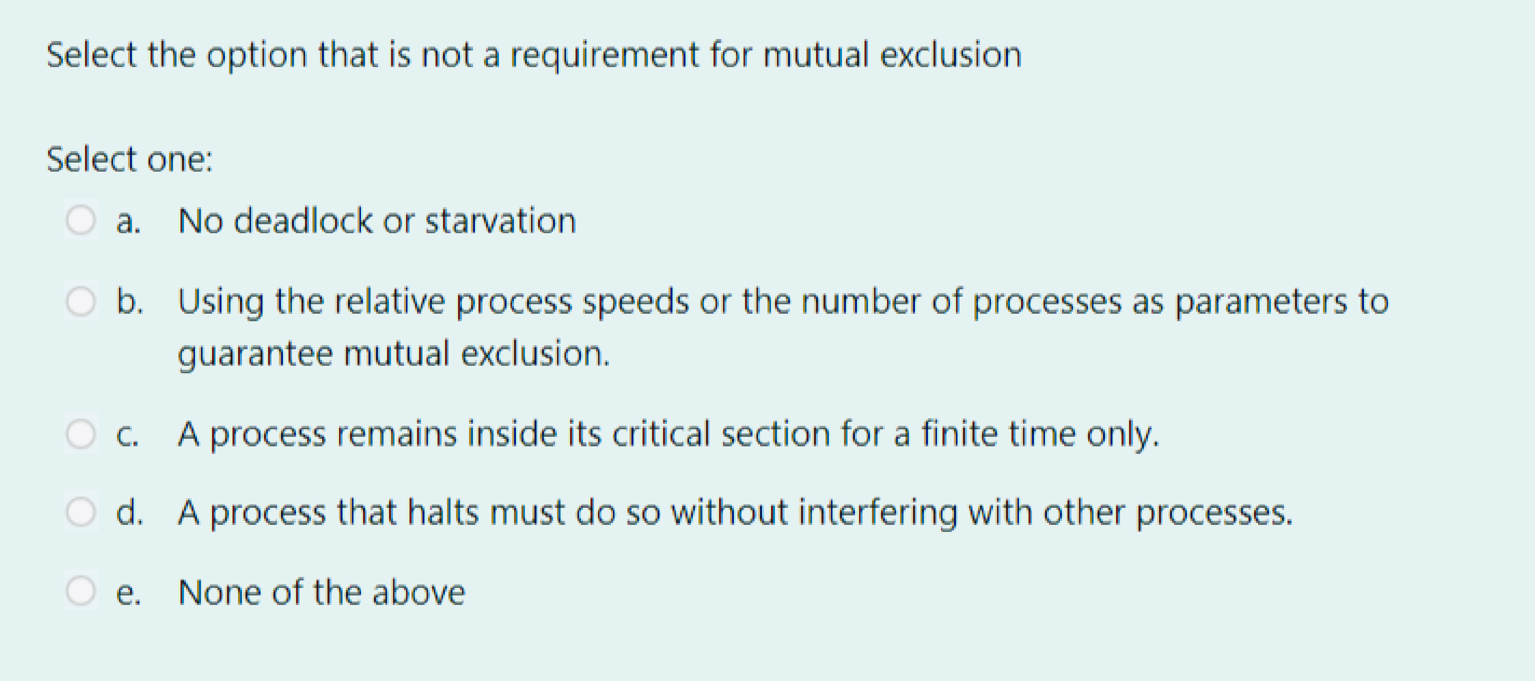
○ a. Requesting all resources at once

○ b. Banker's algorithm

○ c. Detection algorithm

○ d. Ostrich Algorithm

○ e. None of the above



Select the option that is not a requirement for mutual exclusion

Select one:

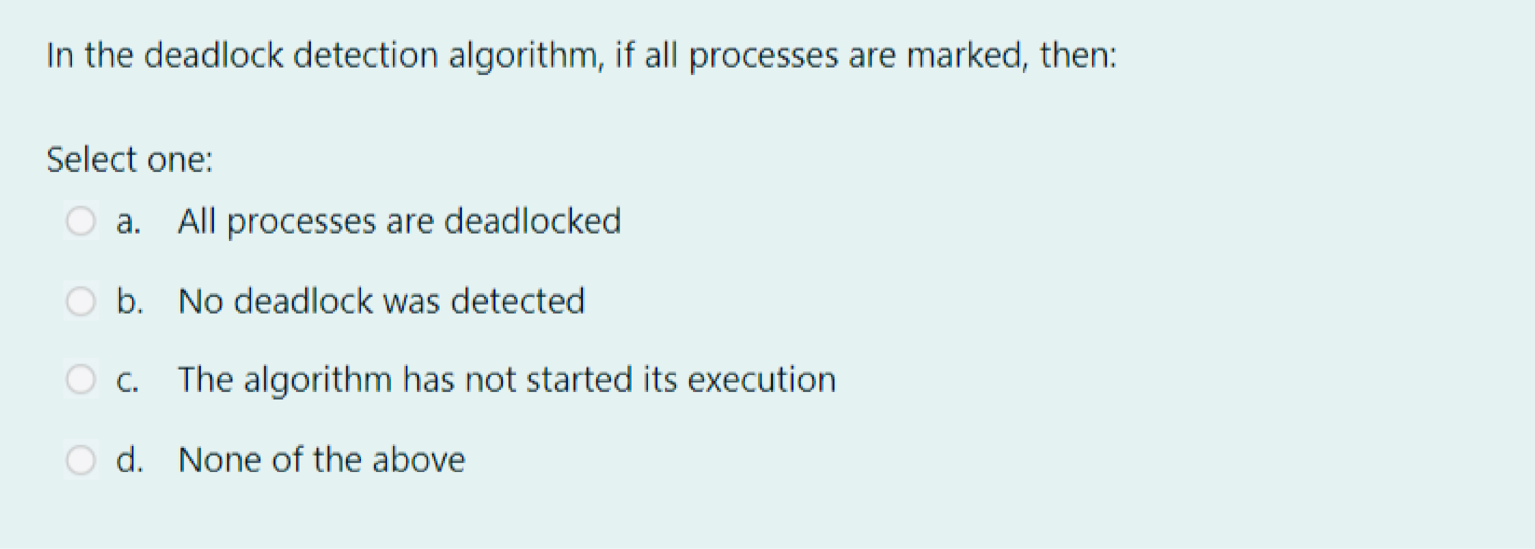
○ a. No deadlock or starvation

○ b. Using the relative process speeds or the number of processes as parameters to guarantee mutual exclusion.

○ c. A process remains inside its critical section for a finite time only.

○ d. A process that halts must do so without interfering with other processes.

○ e. None of the above



In the deadlock detection algorithm, if all processes are marked, then:

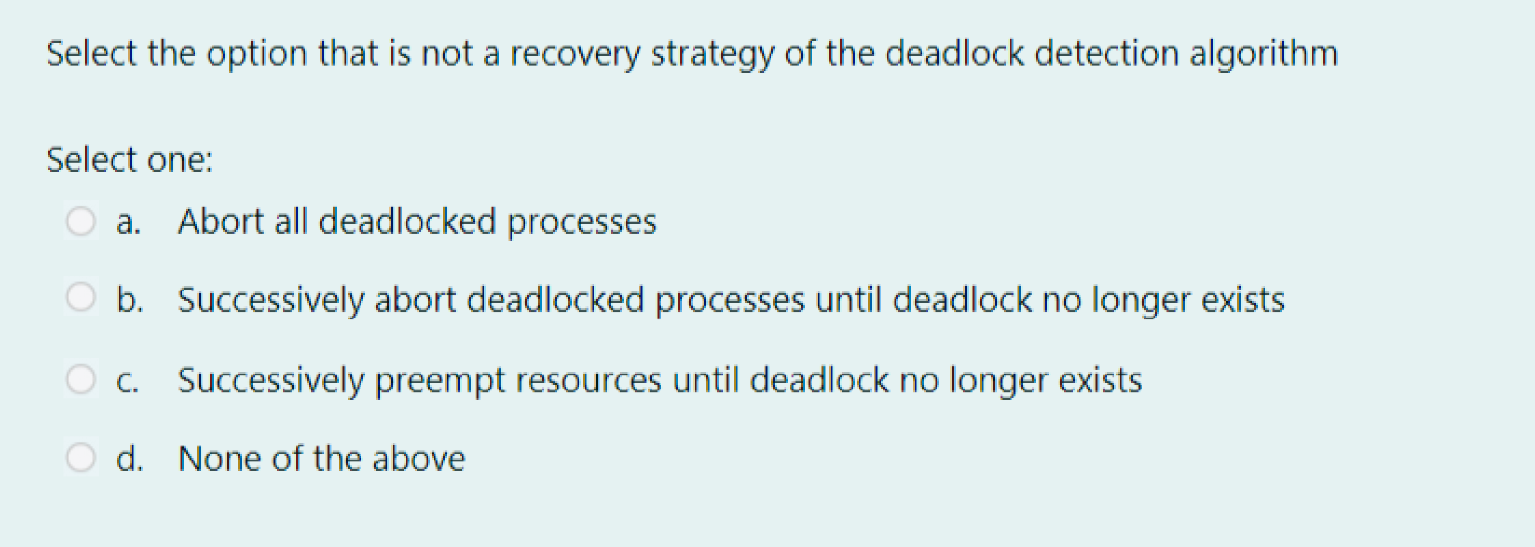
Select one:

○ a. All processes are deadlocked

○ b. No deadlock was detected

○ c. The algorithm has not started its execution

○ d. None of the above



Select the option that is not a recovery strategy of the deadlock detection algorithm

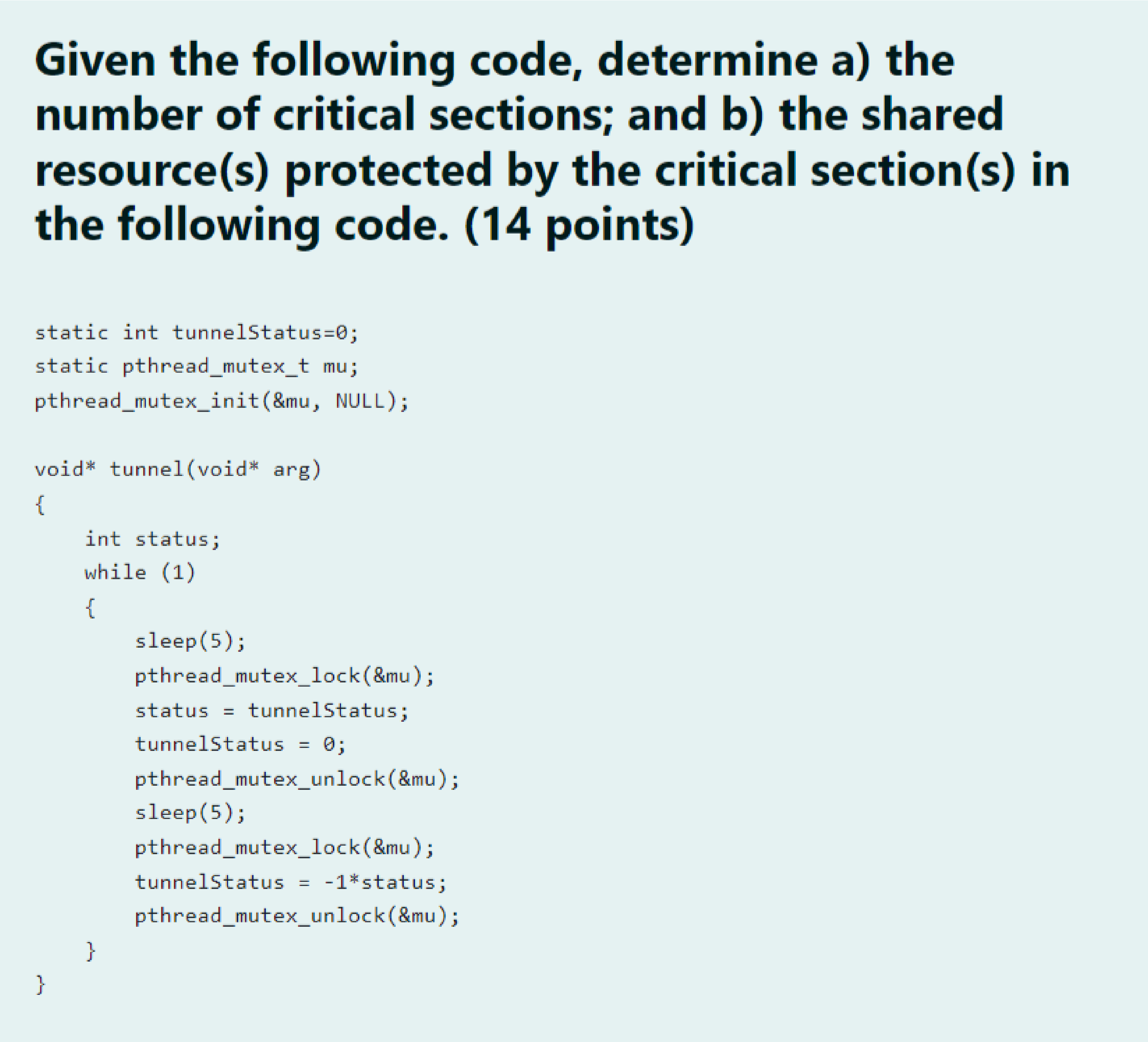
Select one:

○ a. Abort all deadlocked processes

○ b. Successively abort deadlocked processes until deadlock no longer exists

○ c. Successively preempt resources until deadlock no longer exists

○ d. None of the above



Given the following code, determine a) the number of critical sections; and b) the shared resource(s) protected by the critical section(s) in the following code. (14 points)

```
static int tunnelStatus=0;
static pthread_mutex_t mu;
pthread_mutex_init(&mu, NULL);

void* tunnel(void* arg)
{
    int status;
    while (1)
    {
        sleep(5);
        pthread_mutex_lock(&mu);
        status = tunnelStatus;
        tunnelStatus = 0;
        pthread_mutex_unlock(&mu);
        sleep(5);
        pthread_mutex_lock(&mu);
        tunnelStatus = -1*status;
        pthread_mutex_unlock(&mu);
    }
}
```

Number of critical sections

Answer: ☐

Shared Resource(s) controlled by the critical section:

(this is an all-or-nothing question)

Select one or more:

- ☐ a. tunnelStatus
- ☐ b. mu
- ☐ c. status
- ☐ d. sleep(5)