1. Which of the following cannot be shared among different threads of a process?
   a) Process data
   b) Process code
   c) File handles
   d) CPU registers
   e) Stack

2. Shortest Job First Schedulers
   a) ensure fairness
   b) avoid Starvation
   c) minimize average waiting time
   d) both a and b
   e) none of the above

3. Initializing a semaphore S with a negative value \(-N\) may result in a situation where
   a) \(N\) processes will be forced to wait before passing a wait call on \(S\)
   b) if any process is forced to wait, any subsequent signal on \(S\) will restart it
   c) \(N\) processes can pass a wait call before any process is forced to wait
   d) none of the above

4. State the advantages and disadvantages of multi-thread programming over multi-process programming?

5. State the difference between deadlock and starvation?

6. A solution to the critical section problem must satisfy three requirements. Explain briefly the three requirements?
7. What are the possible outputs when the processes A and B run concurrently.
   Assume that the shared variables X and Y are initialized to 0.

<table>
<thead>
<tr>
<th>Process A</th>
<th>Process B</th>
</tr>
</thead>
<tbody>
<tr>
<td>while (X == 0) {</td>
<td>printf(“b”);</td>
</tr>
<tr>
<td>// do nothing</td>
<td>X = 1;</td>
</tr>
<tr>
<td>printf(“a”);</td>
<td>while (Y == 0) {</td>
</tr>
<tr>
<td>Y = 1;</td>
<td>// do nothing</td>
</tr>
<tr>
<td>Y = 0;</td>
<td>}</td>
</tr>
<tr>
<td>printf(“d”);</td>
<td>printf(“c”);</td>
</tr>
<tr>
<td>Y = 1;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Two concurrent processes have the following similar code

```
Process P_i
---------
Top Code
...
Rendezvous point
Bottom Code
...
```

Using semaphores, show by inserting instructions at the rendezvous point, how you can make sure both processes meet at the rendezvous point? In other words, if one of the processes arrives at the rendezvous point earlier than the other, it waits until the other process also arrives there?
What is the minimum number of semaphores needed?
You can declare semaphores of type sem_t. The only operations available to you are sem_init, sem_wait and sem_post.