1.

Gegeben sind drei Prozesse A, B, C mit den folgenden Aufrufen von Semaphor-Operationen.

A	B	C
P(S1) P(S1) P(S1) . . . V(S2) END	P(S2) . . . V(S3) V(S1) END	P(S3) P(S3) P(S3) . . V(S2) V(S2) END

Welche Prozesse erreichen END, wenn die folgenden Initialisierungen (a) - (c) gegen sind.

(a)
```
semainit(S1, 2);
semainit(S2, 0);
semainit(S3, 2);
```

(b)
```
semainit(S1, 3);
semainit(S2, 0);
semainit(S3, 2);
```

(c)
```
semainit(S1, 0);
semainit(S2, 0);
semainit(S3, 3);
```

2. Write a simple pseudocode program using semaphore primitives (P, V) that can get stuck in a deadlock.

3. a) Describe the meaning of the term "pseudo-parallelism". What is "pseudo" about it?
   b) Describe the differences between a preemptive and a non-preemptive scheduling algorithm.

4. What concept allows multiple executions to take place in the same process environment, more or less independently?
   A) interrupts, B) PCBs, C) threads, D) kernel, E) none of these

5. If an allocation algorithm always gives the printer to the process with the smallest print job, what can happen?
   A) circular wait, B) mutual exclusion, C) printer lockup, D) starvation, E) none of these

6. Assume we have three processes A, B, C, which apply for the same ressource, with alphabetical priority. Write a programme in pseudocode, which solves this problem by using general semaphors.