1. **Deadlock Prevention.** Processes may request any combination of the following resources: a cdrom, the soundcard, and the floppy. Describe three different deadlock prevention schemes which denies:

   (a) Hold and Wait

   (b) Circular Wait

   (c) No preemption

2. **Wait-for graph.** Consider the case when process P_0 holds resource R_2 and R_3 and is requesting resources R_4; P_1 holds R_4 and is requesting R_1; and P_2 holds R_1 and is requesting R_3. Draw the wait-for graph. Is the system in deadlock? If so, show what processes constitute the deadlock cycle (Dijkstra's condition 4), and otherwise show the order in which processes could complete.

3. **Safety.**

   (a) Is the following configuration safe? Show your work.

	R_1	R_2	R_3
P_1	1/0	0/1	1/1
P_2	1/1	1/0	0/1
P_3	0/0	0/2	1/0
total resources	2	3	2
available resources	0	2	0

   (b) Is it possible to have an unsafe state in which the safety simulation results in a single process remaining? Explain.

   (c) What is the minimum number of processes and resources necessary to deadlock. Explain why. Give an example.

4. **Banker's algorithm.** Consider the initial state:

	R_1	R_2	R_3	R_4	finished
P_1	0/0	0/0	0/0	0/0	**true**
P_2	0/0	0/0	0/0	0/0	**true**
P_3	0/0	0/0	0/0	0/0	**true**
total resources	2	3	4	3	
available resources	2	3	4	3	

   What happens when the following requests are made:

   (a) P_1 declares $[2, 3, 3, 3]$

   (b) P_2 declares $[1, 1, 2, 1]$

   (c) P_2 requests $[0, 0, 1, 1]$

   (d) P_3 declares $[2, 1, 1, 2]$

(e) P_1 requests $[1, 1, 2, 2]$

(f) P_3 requests $[2, 0, 1, 1]$

(g) P_1 requests $[1, 2, 1, 1]$

5. **Starvation.** Consider a scheduling discipline in which the faster a process executes until blocking, the higher its priority. Can such a discipline cause starvation? Explain.