Ques 8.4

This is probably not a good solution because it yields too large a scope. It is better to define as narrow a scope as possible. This means that if we add another lock to access the synchronization lock of any other resource then it is possible that it would take a lot of time to complete the processes.

Ques 8.6

Argument in favour of installing deadlock avoidance in the system is that we could still ensure that there are no deadlocks. Also, even if the turn-around time is increased, we can still run all the 5000 processes.

Argument against installing deadlock avoidance in the system is that deadlocks occur infrequently and also the cost to rerun the 10 programs is not high. It would not have a major improvement even if we install the deadlock avoidance.

Ques 8.22

The deadlock situation in this condition is not possible as all the three processes take at most 2 resources. Let the situation be that all the three processes require 2 resources. Any one of the processes can first access the required 2 resources while the other two processes are waiting with either 1 or 0 resources. When the process releases the resource either of the two others can access it while the third one may wait or if the resources required are mutually exclusive then both can work together.

Ques 8.24

We can distribute the available n-chopsticks to gif[n/2] (where gif[] is the greatest integer function) philosophers while the rest of the philosophers would be in waiting. And each time one philosopher is done eating another philosopher can take two chopsticks. This means at any time we would not allow philosopher to take only one chopstick. This could prevent the deadlock situation.

Ques 8.28

Part A.

	Allocation	Max	Required	Available
	ABCD	ABCD	ABCD	ABCD
T0	3141	6477	3336	2224
T1	2102	4232	2130	
T2	2413	2533	0120	
T3	4110	6332	2222	
T4	2221	5675	3454	

First, we would execute T3. After the completion the available would become (6,3,3,4). Next, we would execute T2. After the completion the available would become (8,7,4,7). Next, we can execute T0. After the completion the available would become (11,8,8,8). Next, we can execute T1. After the completion the available would become (13,9,8,10). Lastly, we can execute T4. After the completion the available would become (15,11,10,11).

Part B.

	Allocation	Max	Required	Available
	ABCD	ABCD	ABCD	ABCD
T0	3141	6477	3336	2224
T1	2102	4232	2130	
T2	2413	2533	0120	
T3	4110	6332	2222	
T4	2221	5675	3454	

If T4 requests for (2,2,2,4) the availability become (0,0,0,0), allocated becomes (4,4,4,5) and the required resource becomes (1,2,3,0). Hence we arrive at a deadlock situation since none of the threads can proceed further due to lack of resources.

Part C.

	Allocation	Max	Required	Available
	ABCD	ABCD	ABCD	ABCD
T0	3141	6477	3336	2224
T1	2102	4232	2130	
T2	2413	2533	0120	
T3	4110	6332	2222	
T4	2221	5675	3454	

If T2 requests for (0,1,1,0) the availability become (2,1,1,4), allocated becomes (2,5,2,3) and the required resource becomes (0,0,1,0).

	Allocation	Max	Required	Available
	ABCD	ABCD	ABCD	ABCD
T0	3141	6477	3336	0010
T1	2102	4232	2130	
T2	2523	2533	0010	
T3	4110	6332	2222	
T4	2221	5675	3454	

First, we can execute T2. After the completion the available would become (4,6,3,7). Next, we can execute T0. After the completion the available would become (7,7,7,8). Next, we can execute T1. After the completion the available would become (9,8,7,10). Next, we can execute T3. After the completion the available would become (13,9,8,10). Lastly, we can execute T4. After the completion the available would become (15,11,10,11).

Hence, we can grant the resources immediately to T2.

Part D.

	Allocation	Max	Required	Available
	ABCD	ABCD	ABCD	ABCD
T0	3141	6477	3336	2224
T1	2102	4232	2130	
T2	2413	2533	0120	
T3	4110	6332	2222	
T4	2221	5675	3454	

If T3 requests for (2,2,1,2) the availability become (0,0,1,2), allocated becomes (6,3,2,2) and the required resource becomes (0,0,1,0).

	Allocation	Max	Required	Available
	ABCD	ABCD	ABCD	ABCD
T0	3141	6477	3336	0012
T1	2102	4232	2130	
T2	2413	2533	0120	
T3	6322	6332	0010	
T4	2221	5675	3454	

First, we can execute T3. After the completion the available would become (6,3,3,4). Next, we can execute T1. After the completion the available would become (8,4,3,6). Next, we can execute T0. After the completion the available would become (11,5,7,7). Next, we can execute T2. After the completion the available would become (13,9,8,10). Lastly, we can execute T4. After the completion the available would become (15,11,10,11).

Hence, we can grant the resources immediately to T3.