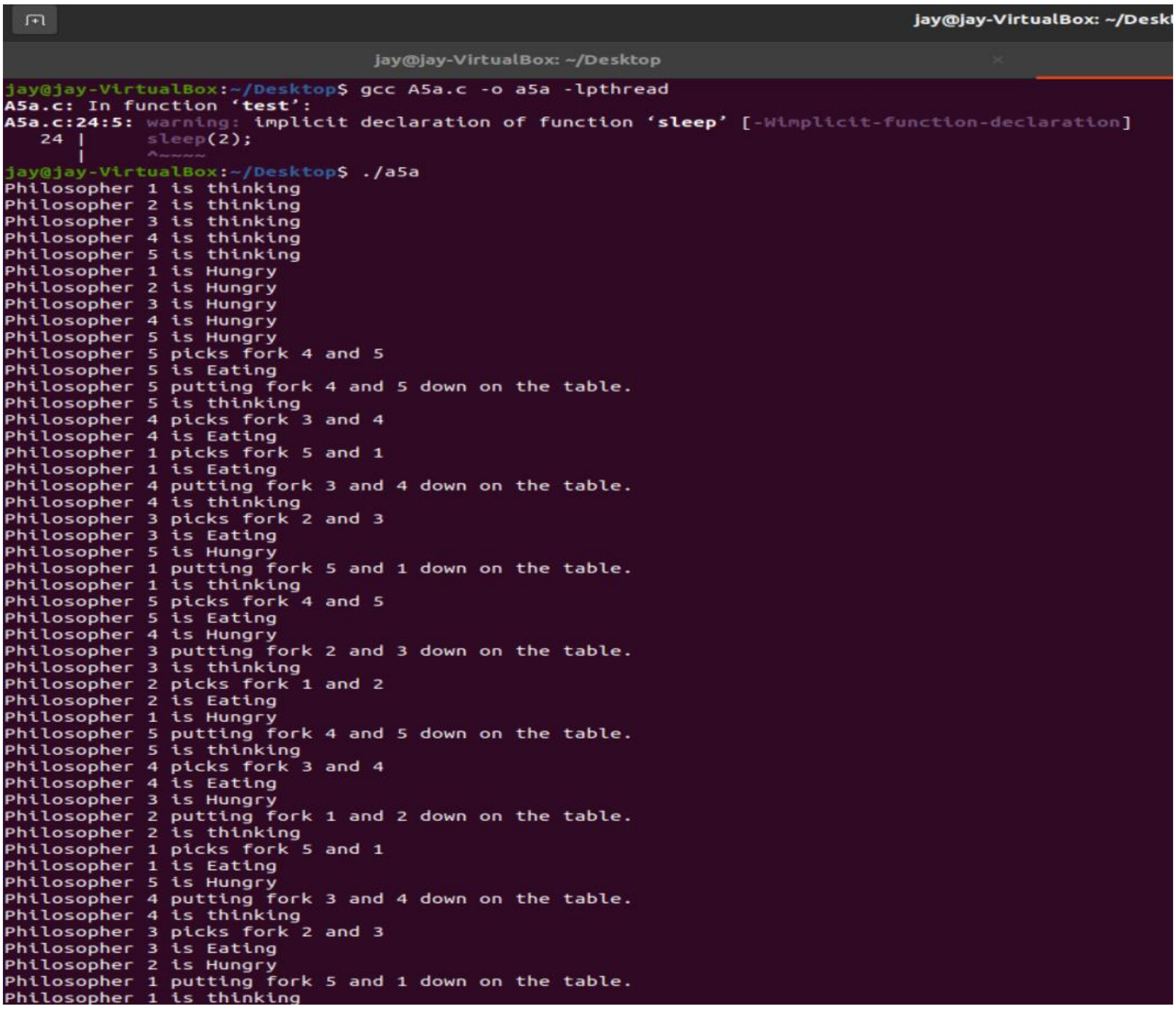
ASSIGNMENT 5 (BONUS ASSIGNMENT)

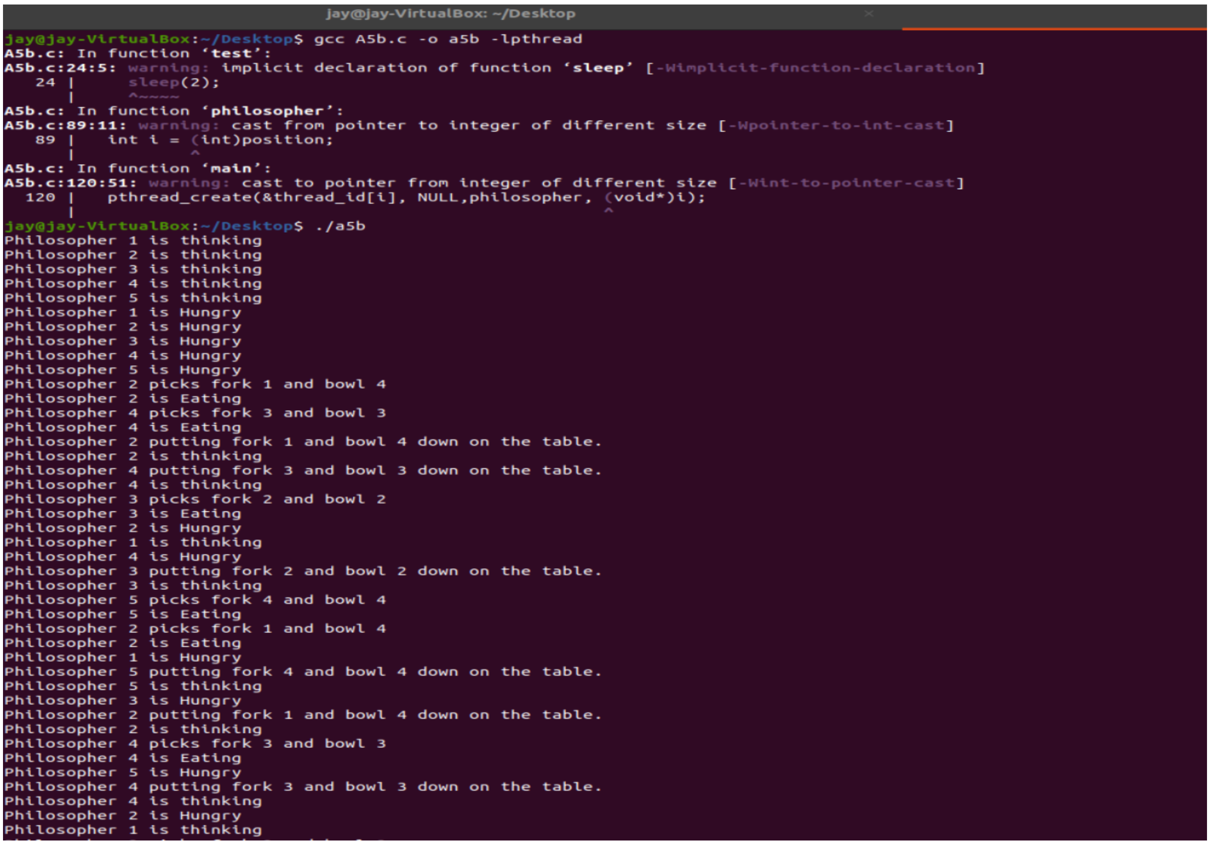
(JAY SARAF 2020438)

1. a) There will be a deadlock condition.
   There are two semaphores being used for the dining philosopher problem. Mutex semaphore is being used to prevent picking up or putting down of the same fork by two different philosophers at the same time. Mutex is a locking mechanism which prevents multiple threads from acquiring mutex at the same time and ensure entry to the critical section. Philosopher checks whether both the forks on his right and left are free and then he picks both of them.



```
jay@jay-VirtualBox:~/Desktop$ gcc A5a.c -o a5a -lpthread
A5a.c: In function ‘test’:
A5a.c:24:5: warning: implicit declaration of function ‘sleep’ [-Wimplicit-function-declaration]
   24 |     sleep(2);
      |     ^~~~~
jay@jay-VirtualBox:~/Desktop$ ./a5a
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 5 is thinking
Philosopher 1 is Hungry
Philosopher 2 is Hungry
Philosopher 3 is Hungry
Philosopher 4 is Hungry
Philosopher 5 is Hungry
Philosopher 5 picks fork 4 and 5
Philosopher 5 is Eating
Philosopher 5 putting fork 4 and 5 down on the table.
Philosopher 5 is thinking
Philosopher 4 picks fork 3 and 4
Philosopher 4 is Eating
Philosopher 1 picks fork 5 and 1
Philosopher 1 is Eating
Philosopher 4 putting fork 3 and 4 down on the table.
Philosopher 4 is thinking
Philosopher 3 picks fork 2 and 3
Philosopher 3 is Eating
Philosopher 5 is Hungry
Philosopher 1 putting fork 5 and 1 down on the table.
Philosopher 1 is thinking
Philosopher 5 picks fork 4 and 5
Philosopher 5 is Eating
Philosopher 4 is Hungry
Philosopher 3 putting fork 2 and 3 down on the table.
Philosopher 3 is thinking
Philosopher 2 picks fork 1 and 2
Philosopher 2 is Eating
Philosopher 1 is Hungry
Philosopher 5 putting fork 4 and 5 down on the table.
Philosopher 5 is thinking
Philosopher 4 picks fork 3 and 4
Philosopher 4 is Eating
Philosopher 3 is Hungry
Philosopher 2 putting fork 1 and 2 down on the table.
Philosopher 2 is thinking
Philosopher 1 picks fork 5 and 1
Philosopher 1 is Eating
Philosopher 5 is Hungry
Philosopher 4 putting fork 3 and 4 down on the table.
Philosopher 4 is thinking
Philosopher 3 picks fork 2 and 3
Philosopher 3 is Eating
Philosopher 2 is Hungry
Philosopher 1 putting fork 5 and 1 down on the table.
Philosopher 1 is thinking
```

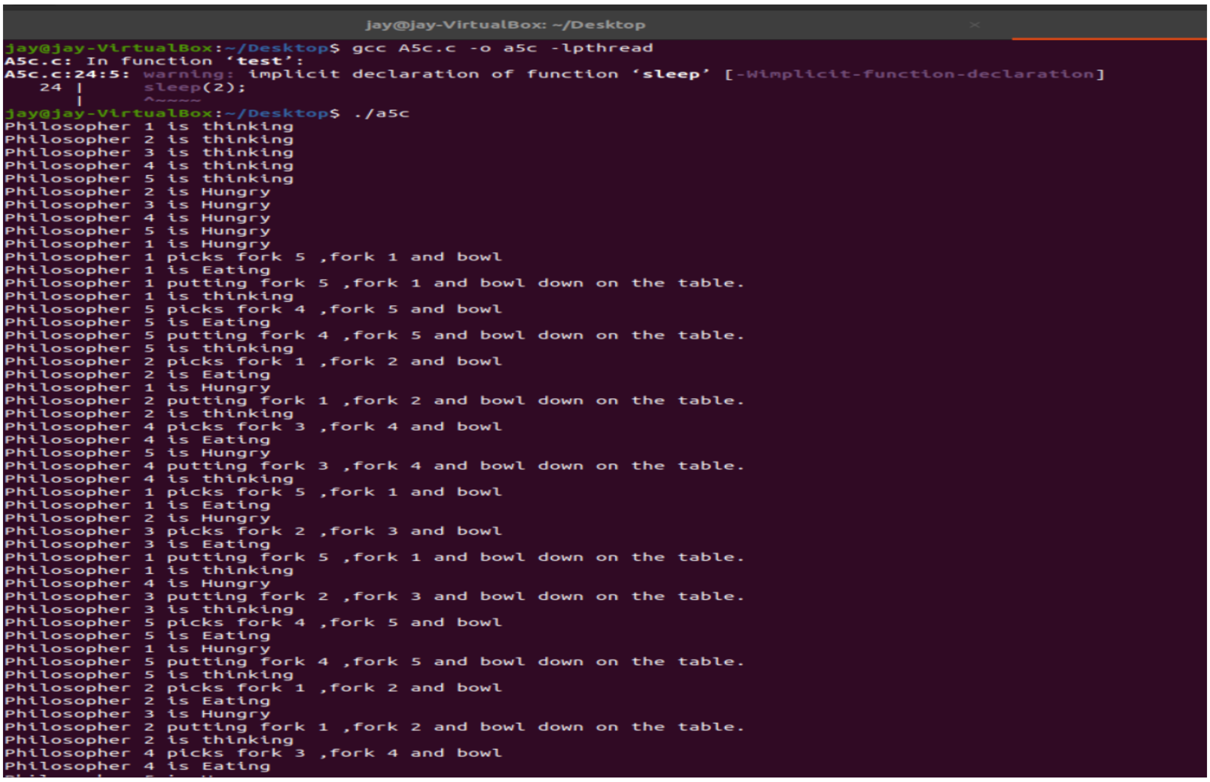
b) There can be a deadlock. As one philosopher can use only one fork at a time for eating so there won’t be any starvation, deadlock or problem in resource allocation in terms of fork. But as there are only four bowls so bowls will determine the process. If bowl gets interchanged between only 4 philosophers, then the 5th philosopher won’t be able to eat creating a deadlock situation for the 5th philosopher. Semaphore mutex is initialized by 4 as there are four bowls. Philosopher checks whether there is a fork on his left.



```
jay@jay-VirtualBox:~/Desktop$ gcc A5b.c -o a5b -lpthread
A5b.c: In function ‘test’:
A5b.c:24:5: warning: implicit declaration of function ‘sleep’ [-Wimplicit-function-declaration]
   24 |     sleep(2);
      |     ^~~~~
A5b.c: In function ‘philosopher’:
A5b.c:89:11: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
   89 |   int i = (int)position;
      |           ^
A5b.c: In function ‘main’:
A5b.c:120:51: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
  120 |   pthread_create(&thread_id[i], NULL,philosopher, (void*)i);
      |                                                   ^
jay@jay-VirtualBox:~/Desktop$ ./a5b
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 5 is thinking
Philosopher 1 is Hungry
Philosopher 2 is Hungry
Philosopher 3 is Hungry
Philosopher 4 is Hungry
Philosopher 5 is Hungry
Philosopher 2 picks fork 1 and bowl 4
Philosopher 2 is Eating
Philosopher 4 picks fork 3 and bowl 3
Philosopher 4 is Eating
Philosopher 2 putting fork 1 and bowl 4 down on the table.
Philosopher 2 is thinking
Philosopher 4 putting fork 3 and bowl 3 down on the table.
Philosopher 4 is thinking
Philosopher 3 picks fork 2 and bowl 2
Philosopher 3 is Eating
Philosopher 2 is Hungry
Philosopher 1 is thinking
Philosopher 4 is Hungry
Philosopher 3 putting fork 2 and bowl 2 down on the table.
Philosopher 3 is thinking
Philosopher 5 picks fork 4 and bowl 4
Philosopher 5 is Eating
Philosopher 2 picks fork 1 and bowl 4
Philosopher 2 is Eating
Philosopher 1 is Hungry
Philosopher 5 putting fork 4 and bowl 4 down on the table.
Philosopher 5 is thinking
Philosopher 3 is Hungry
Philosopher 2 putting fork 1 and bowl 4 down on the table.
Philosopher 2 is thinking
Philosopher 4 picks fork 3 and bowl 3
Philosopher 4 is Eating
Philosopher 5 is Hungry
Philosopher 4 putting fork 3 and bowl 3 down on the table.
Philosopher 4 is thinking
Philosopher 2 is Hungry
Philosopher 1 is thinking
```

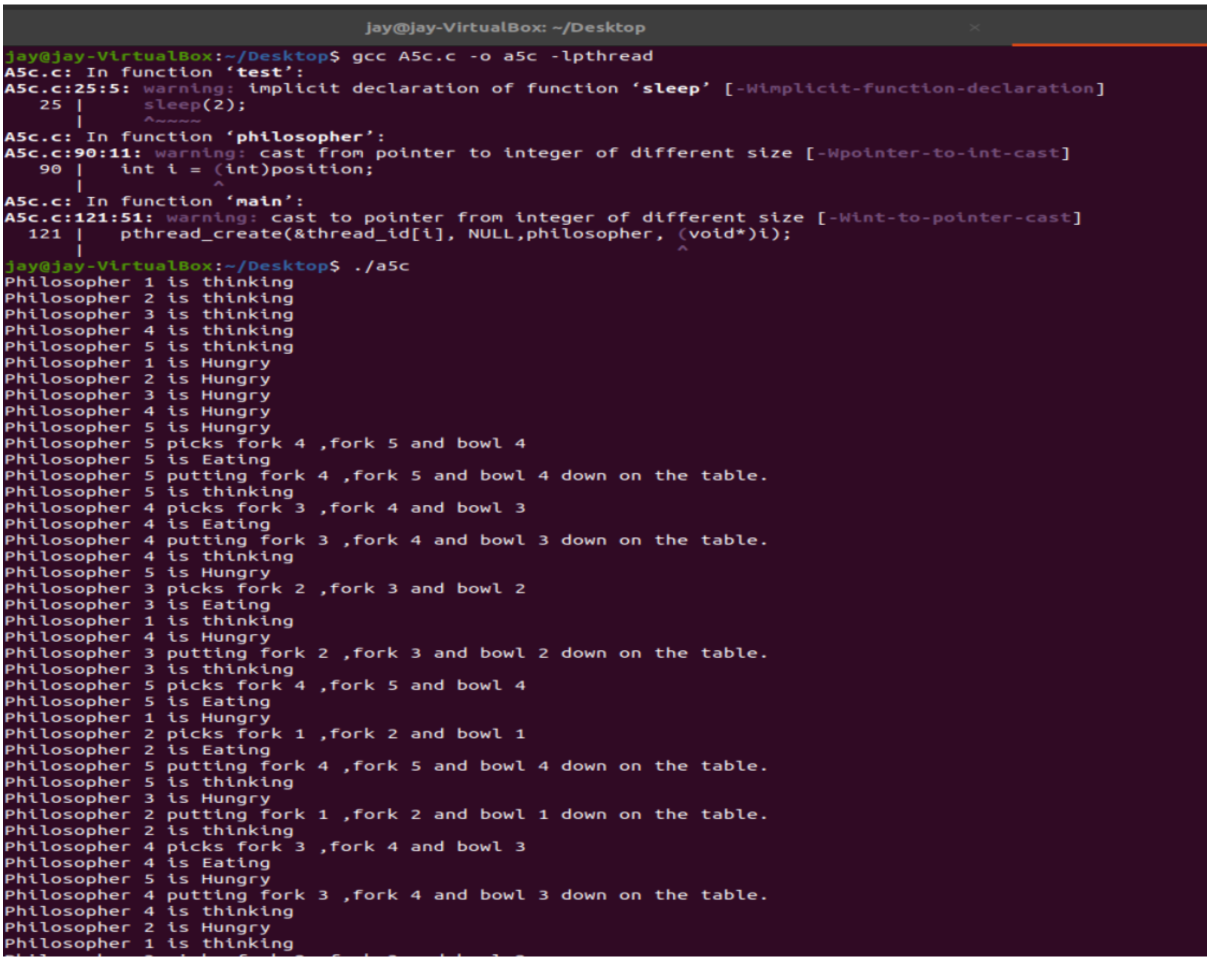
c) There will be a deadlock condition. But the deadlock condition won’t depend on the bowl, it would depend upon the pair of forks being used. Semaphore mutex is initialized by 4 as there are four bowls.

Without bowl numbering.



```
jay@jay-VirtualBox:~/Desktop$ gcc A5c.c -o a5c -lpthread
A5c.c: In function ‘test’:
A5c.c:24:5: warning: implicit declaration of function ‘sleep’ [-Wimplicit-function-declaration]
   24 |     sleep(2);
      |     ^~~~~
jay@jay-VirtualBox:~/Desktop$ ./a5c
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 5 is thinking
Philosopher 2 is Hungry
Philosopher 3 is Hungry
Philosopher 4 is Hungry
Philosopher 5 is Hungry
Philosopher 1 is Hungry
Philosopher 1 picks fork 5 ,fork 1 and bowl
Philosopher 1 is Eating
Philosopher 1 putting fork 5 ,fork 1 and bowl down on the table.
Philosopher 1 is thinking
Philosopher 5 picks fork 4 ,fork 5 and bowl
Philosopher 5 is Eating
Philosopher 5 putting fork 4 ,fork 5 and bowl down on the table.
Philosopher 5 is thinking
Philosopher 2 picks fork 1 ,fork 2 and bowl
Philosopher 2 is Eating
Philosopher 1 is Hungry
Philosopher 2 putting fork 1 ,fork 2 and bowl down on the table.
Philosopher 2 is thinking
Philosopher 4 picks fork 3 ,fork 4 and bowl
Philosopher 4 is Eating
Philosopher 5 is Hungry
Philosopher 4 putting fork 3 ,fork 4 and bowl down on the table.
Philosopher 4 is thinking
Philosopher 1 picks fork 5 ,fork 1 and bowl
Philosopher 1 is Eating
Philosopher 2 is Hungry
Philosopher 3 picks fork 2 ,fork 3 and bowl
Philosopher 3 is Eating
Philosopher 1 putting fork 5 ,fork 1 and bowl down on the table.
Philosopher 1 is thinking
Philosopher 4 is Hungry
Philosopher 3 putting fork 2 ,fork 3 and bowl down on the table.
Philosopher 3 is thinking
Philosopher 5 picks fork 4 ,fork 5 and bowl
Philosopher 5 is Eating
Philosopher 1 is Hungry
Philosopher 5 putting fork 4 ,fork 5 and bowl down on the table.
Philosopher 5 is thinking
Philosopher 2 picks fork 1 ,fork 2 and bowl
Philosopher 2 is Eating
Philosopher 3 is Hungry
Philosopher 2 putting fork 1 ,fork 2 and bowl down on the table.
Philosopher 2 is thinking
Philosopher 4 picks fork 3 ,fork 4 and bowl
Philosopher 4 is Eating
```

With bowl numbering.



```
jay@jay-VirtualBox:~/Desktop$ gcc A5c.c -o a5c -lpthread
A5c.c: In function ‘test’:
A5c.c:25:5: warning: implicit declaration of function ‘sleep’ [-Wimplicit-function-declaration]
   25 |     sleep(2);
      |     ^~~~~
A5c.c: In function ‘philosopher’:
A5c.c:90:11: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
   90 |   int i = (int)position;
      |           ^
A5c.c: In function ‘main’:
A5c.c:121:51: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
  121 |   pthread_create(&thread_id[i], NULL,philosopher, (void*)i);
      |                                                   ^
jay@jay-VirtualBox:~/Desktop$ ./a5c
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 5 is thinking
Philosopher 1 is Hungry
Philosopher 2 is Hungry
Philosopher 3 is Hungry
Philosopher 4 is Hungry
Philosopher 5 is Hungry
Philosopher 5 picks fork 4 ,fork 5 and bowl 4
Philosopher 5 is Eating
Philosopher 5 putting fork 4 ,fork 5 and bowl 4 down on the table.
Philosopher 5 is thinking
Philosopher 4 picks fork 3 ,fork 4 and bowl 3
Philosopher 4 is Eating
Philosopher 4 putting fork 3 ,fork 4 and bowl 3 down on the table.
Philosopher 4 is thinking
Philosopher 5 is Hungry
Philosopher 3 picks fork 2 ,fork 3 and bowl 2
Philosopher 3 is Eating
Philosopher 1 is thinking
Philosopher 4 is Hungry
Philosopher 3 putting fork 2 ,fork 3 and bowl 2 down on the table.
Philosopher 3 is thinking
Philosopher 5 picks fork 4 ,fork 5 and bowl 4
Philosopher 5 is Eating
Philosopher 1 is Hungry
Philosopher 2 picks fork 1 ,fork 2 and bowl 1
Philosopher 2 is Eating
Philosopher 5 putting fork 4 ,fork 5 and bowl 4 down on the table.
Philosopher 5 is thinking
Philosopher 3 is Hungry
Philosopher 2 putting fork 1 ,fork 2 and bowl 1 down on the table.
Philosopher 2 is thinking
Philosopher 4 picks fork 3 ,fork 4 and bowl 3
Philosopher 4 is Eating
Philosopher 5 is Hungry
Philosopher 4 putting fork 3 ,fork 4 and bowl 3 down on the table.
Philosopher 4 is thinking
Philosopher 2 is Hungry
Philosopher 1 is thinking
```