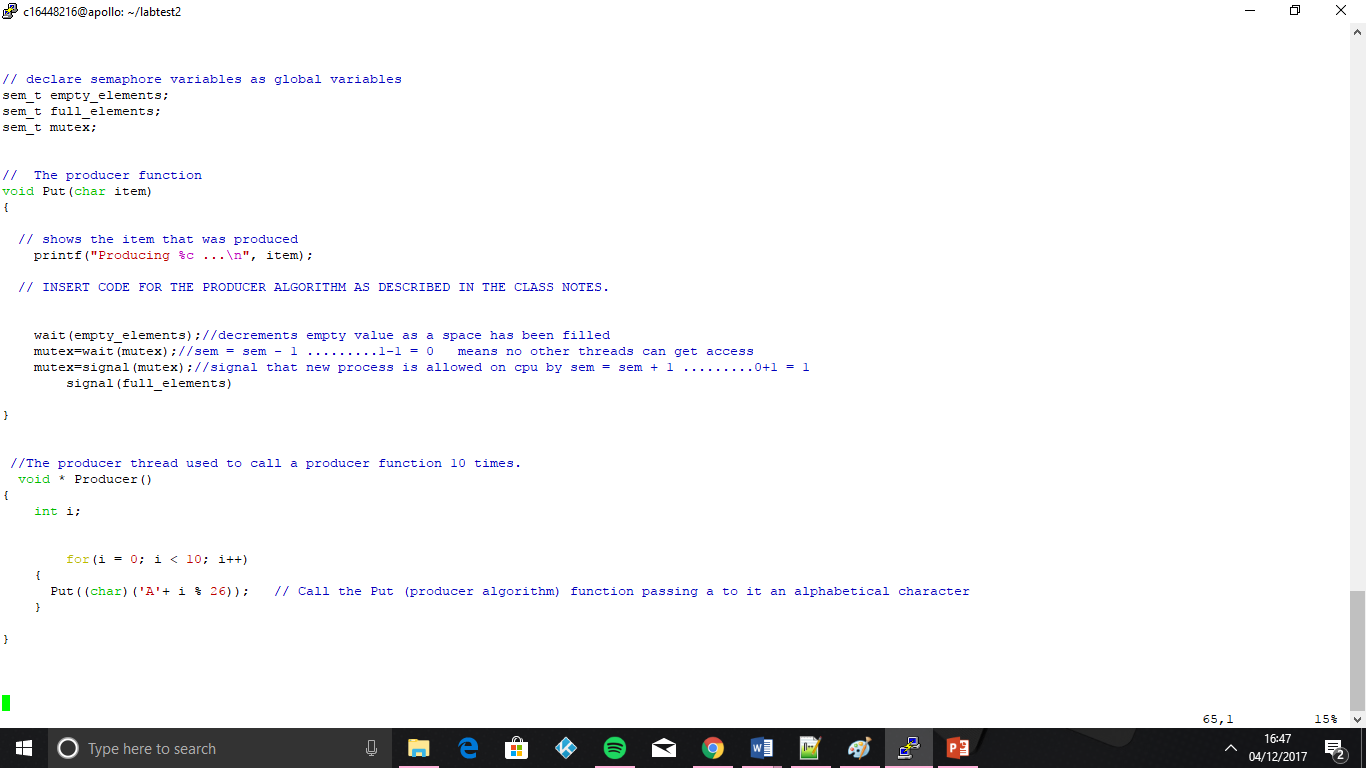
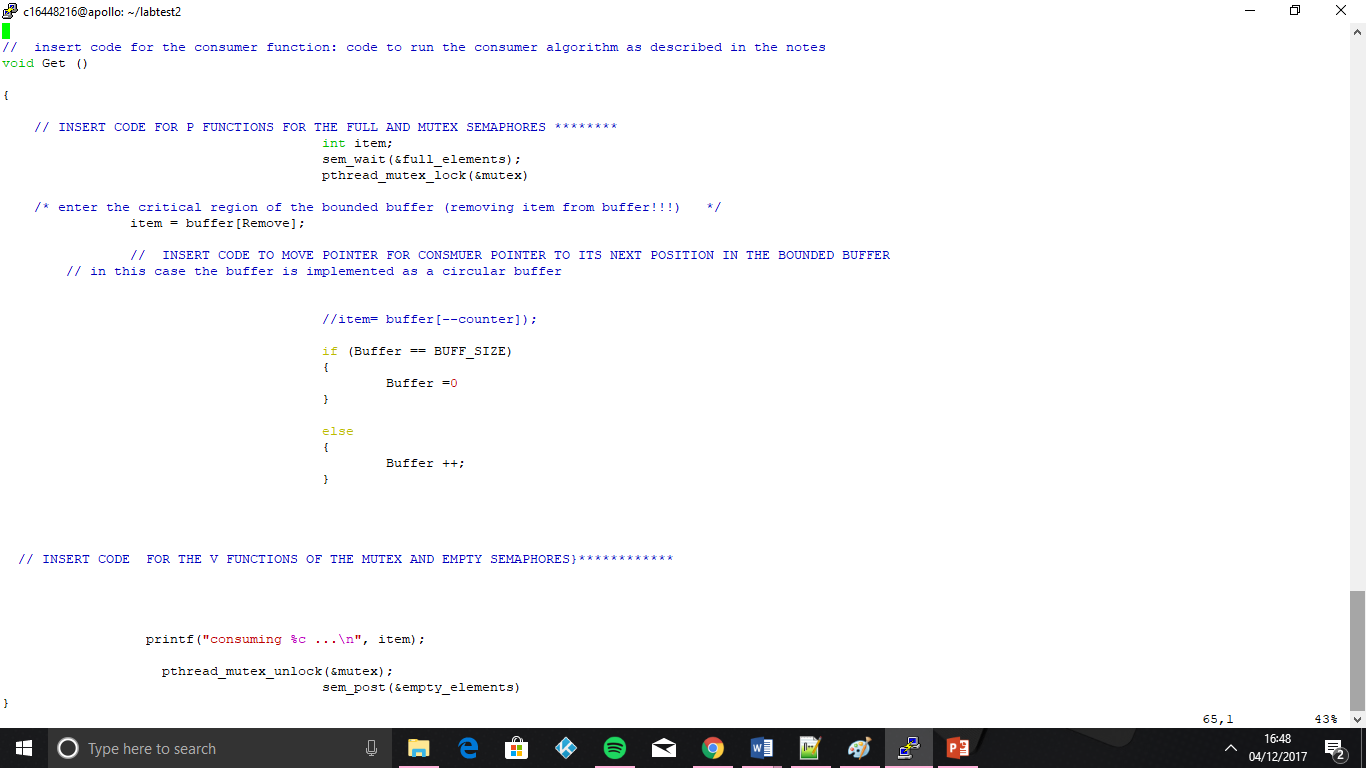
Lab Test 2

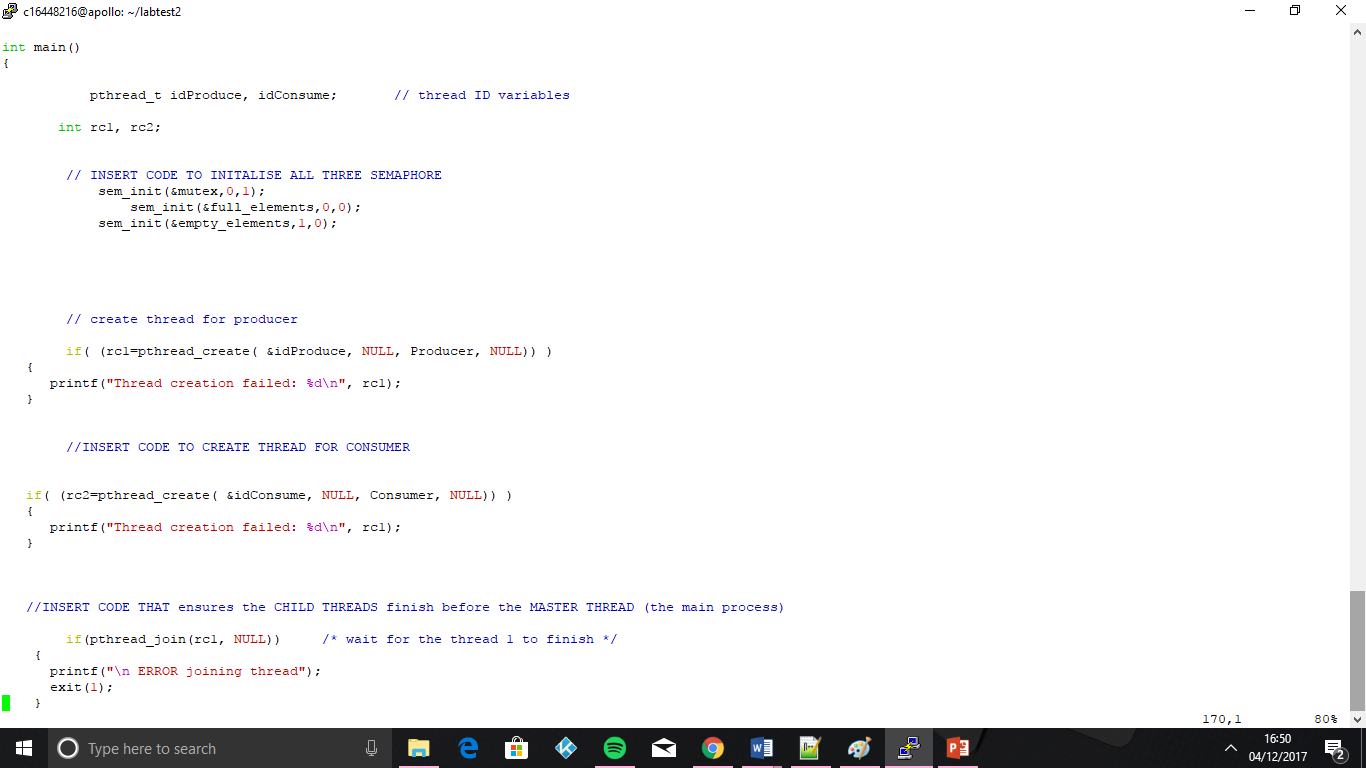
Student Number: C16448216

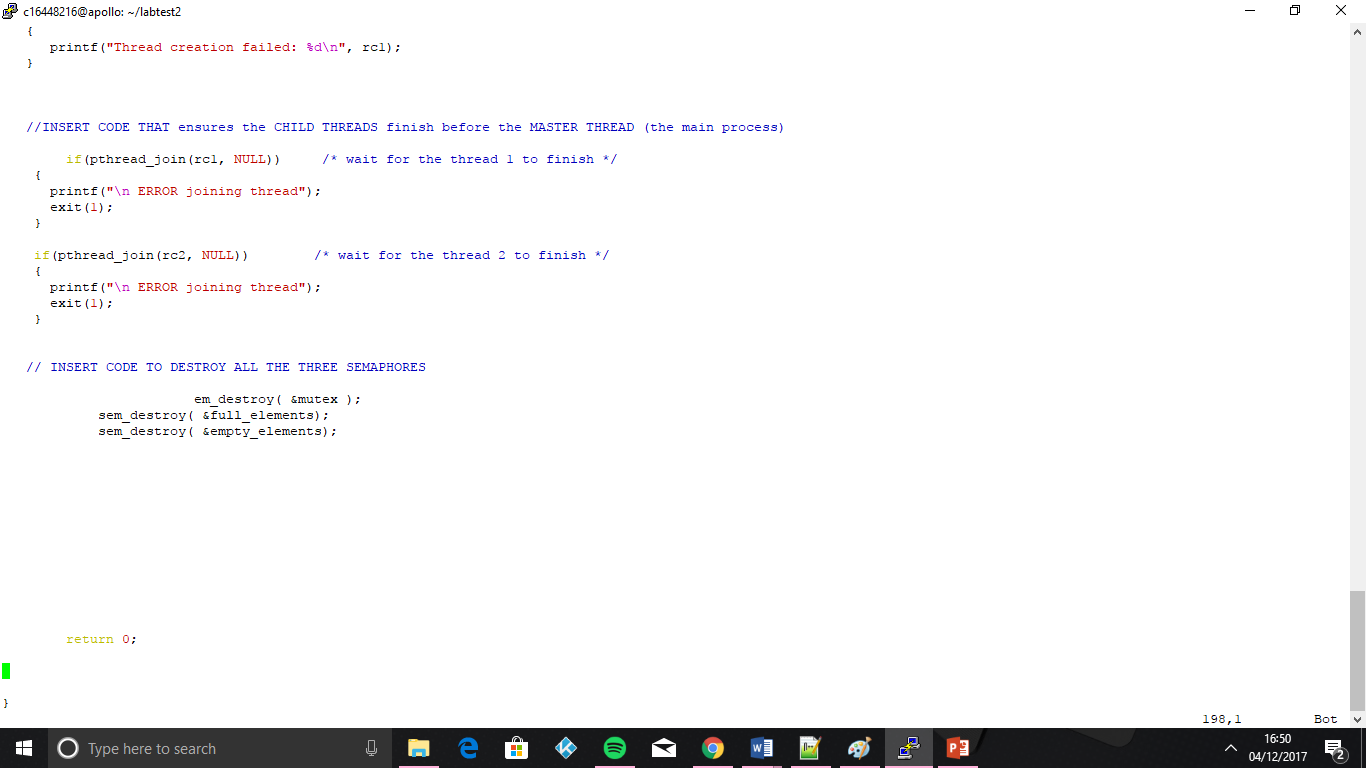
Student Name: Adam Noone

Question1

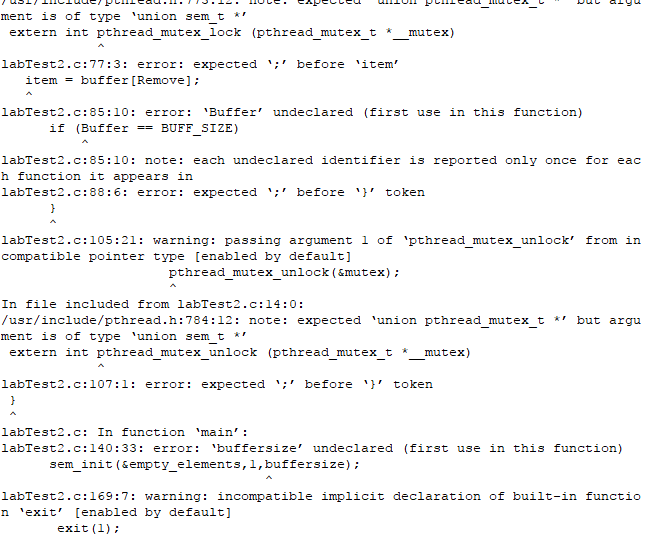






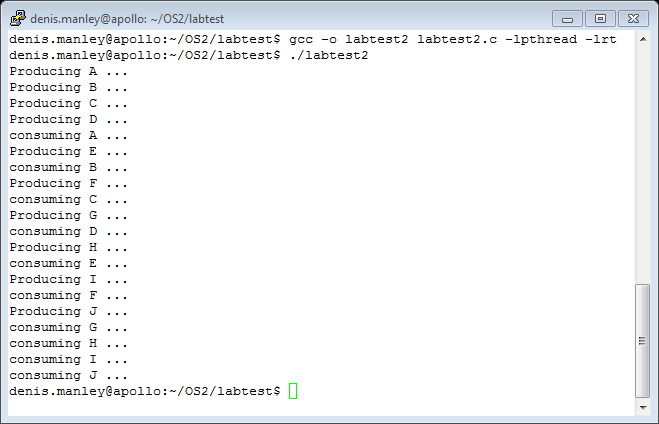


Question2



Question 3

The student must give *an explanation* of the output or expected output (if not completed) in the **LabTest 2** document. This must clearly describe how **semaphores** ensures that the bounded buffer will not allow the *producer function* to add items to a buffer that is full and will not allow the *consumer function* to remove elements from an empty buffer.

The expected output is 

The buffer is at this point empty, so we can call the producer(the producer will check If the buffer is full,if it is full it will wait on the consumer to consume) . this decrements the semaphore , it produces A ,then signals to the consumer function that buffer is not empty by incrementing the semaphore.

At this point the buffer is still not full so we can call the producer again, it produces B.

As the buffer is still not full we are allowed to call the producer 2 more times, creating C and D.

Next, we call the Consumer. The consumer checks that the buffer is not empty (if it is empty it waits for the producer to produce)As the buffer is not empty the at this point because it contains A,B,C,D ,the consumes the block at which its pointing (consumes A) and then points to next block.

The buffer is still not full so we can produce E

Consumer is called and it consumes the block , B

The buffer is still not full so we can produce F

Consumer is called and it consumes the block , C

The buffer is still not full so we can produce G

Consumer is called and it consumes the block , D

The buffer is still not full so we can produce H

Consumer is called and it consumes the block , E

The buffer is still not full so we can produce I

Consumer is called and it consumes the block , F

The buffer is still not full so we can produce J

Consumer is called and it consumes the block , G

Now the program stops reading to the buffer(producing), so the consumer is called until buffer is once again empty ,Consuming G,H,I,J

In a bounded buffer we have a producer and a consumer . the Producers work is to produce data or items and put in buffer. Consumer work is to remove data from buffer and consume it. We have to make sure that producer does not produce data when buffer is full and consumer do not remove data when buffer is empty.

That is why when we are implementing a bounded buffer we must adhere to the guidelines

-if the buffer is full the producer waits

-if the buffer is empty not allowed call the consumer

This is the case because, you will get deadlock if you call the consumer before the producer. If you call the consumer 1st it will lock the mutex denning access to cpu for all threads. The consumer will set the mutex to 0 causing it to wait & because the mutex is 0 the producer can’t write to the buffer, so you get deadlock. that’s why we check if buffer is empty before we run consumer