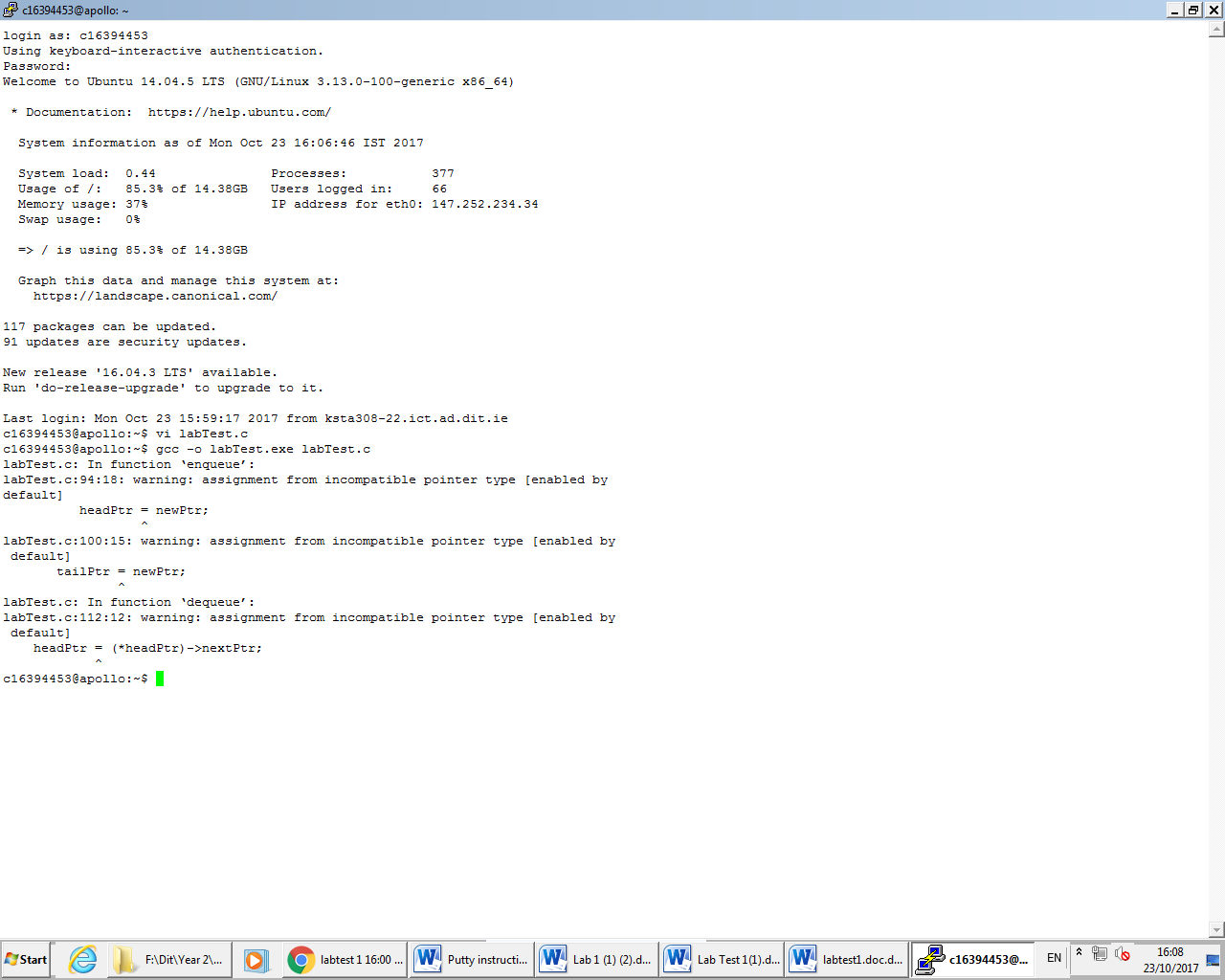
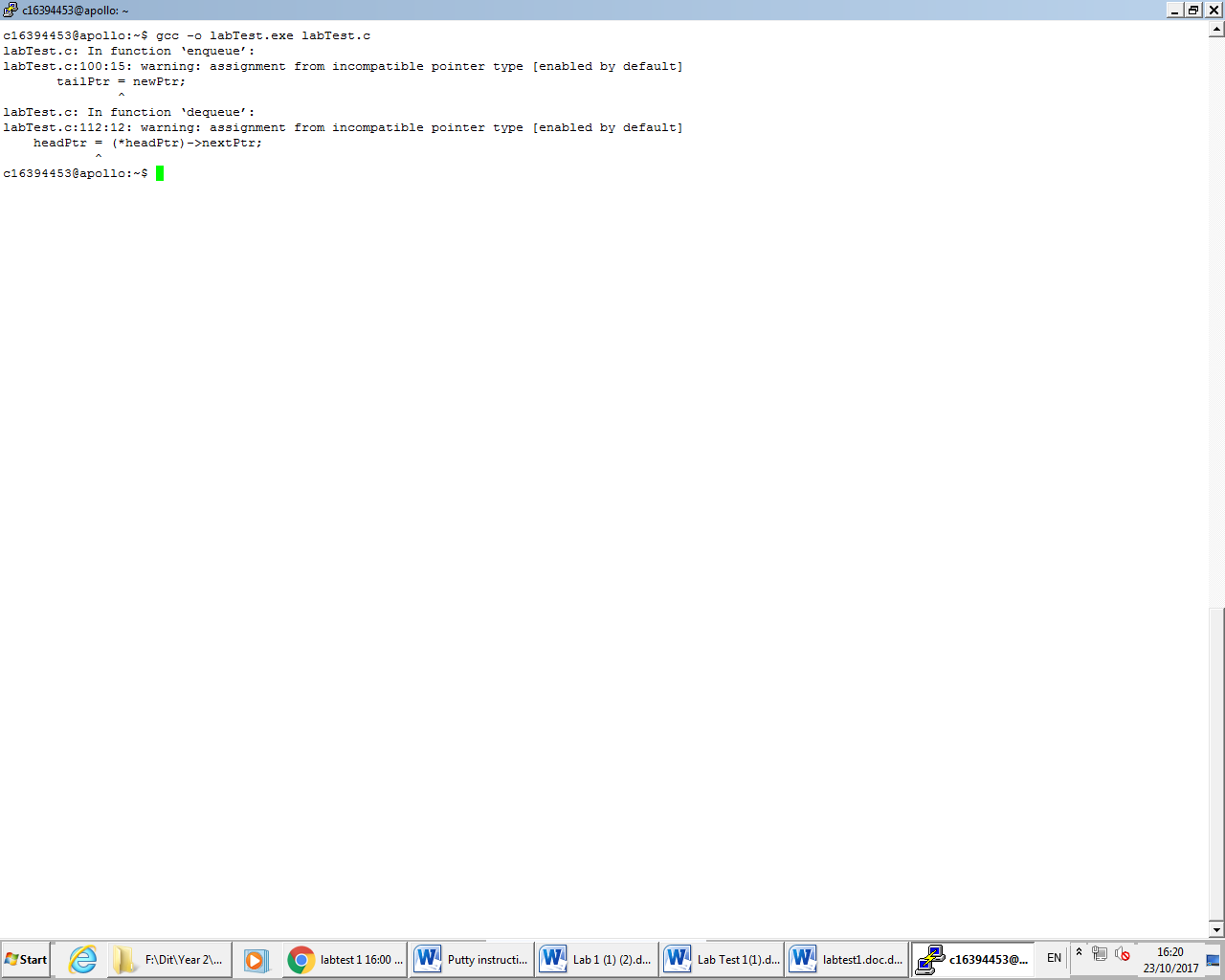
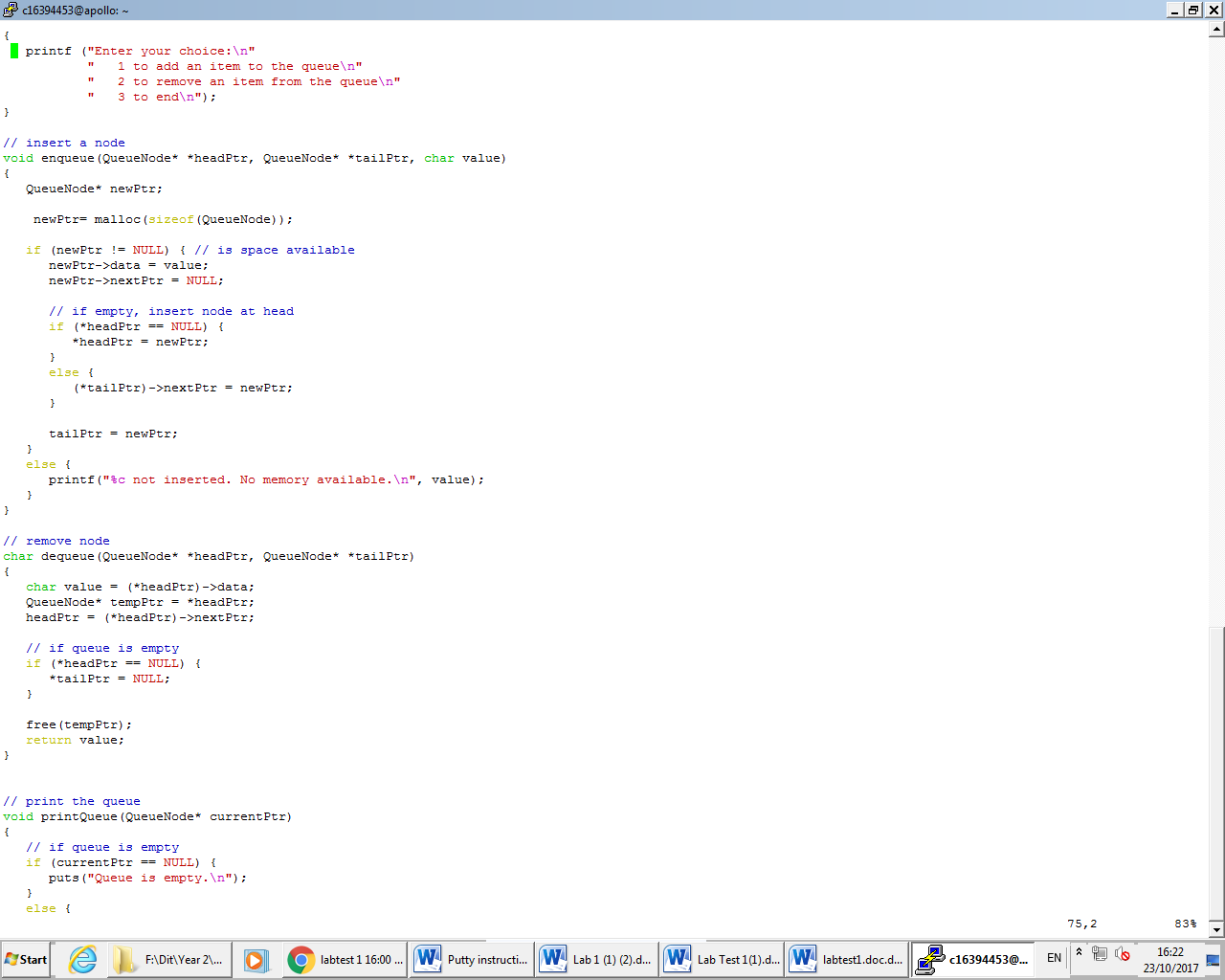
Thomas Killeen C16394453

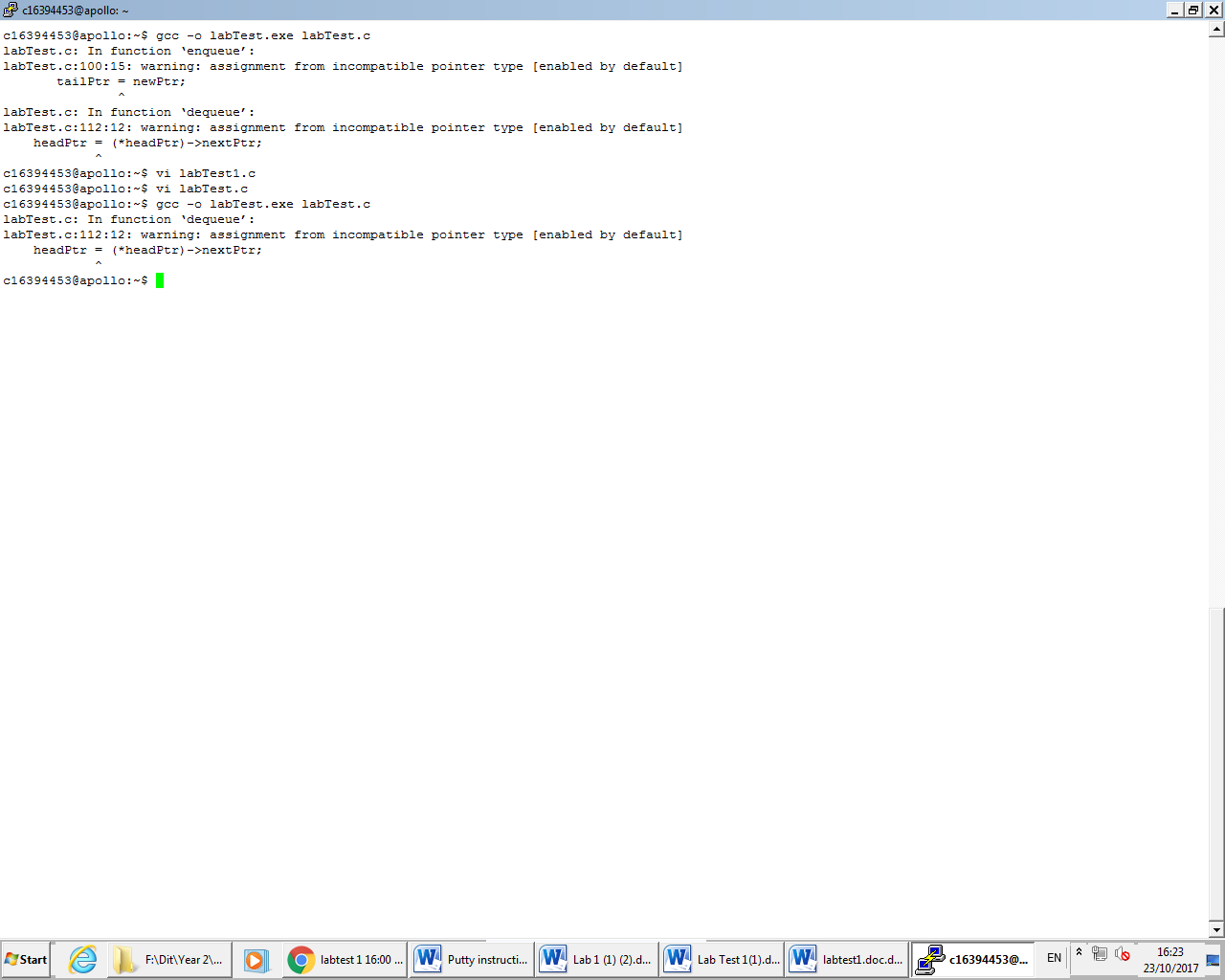


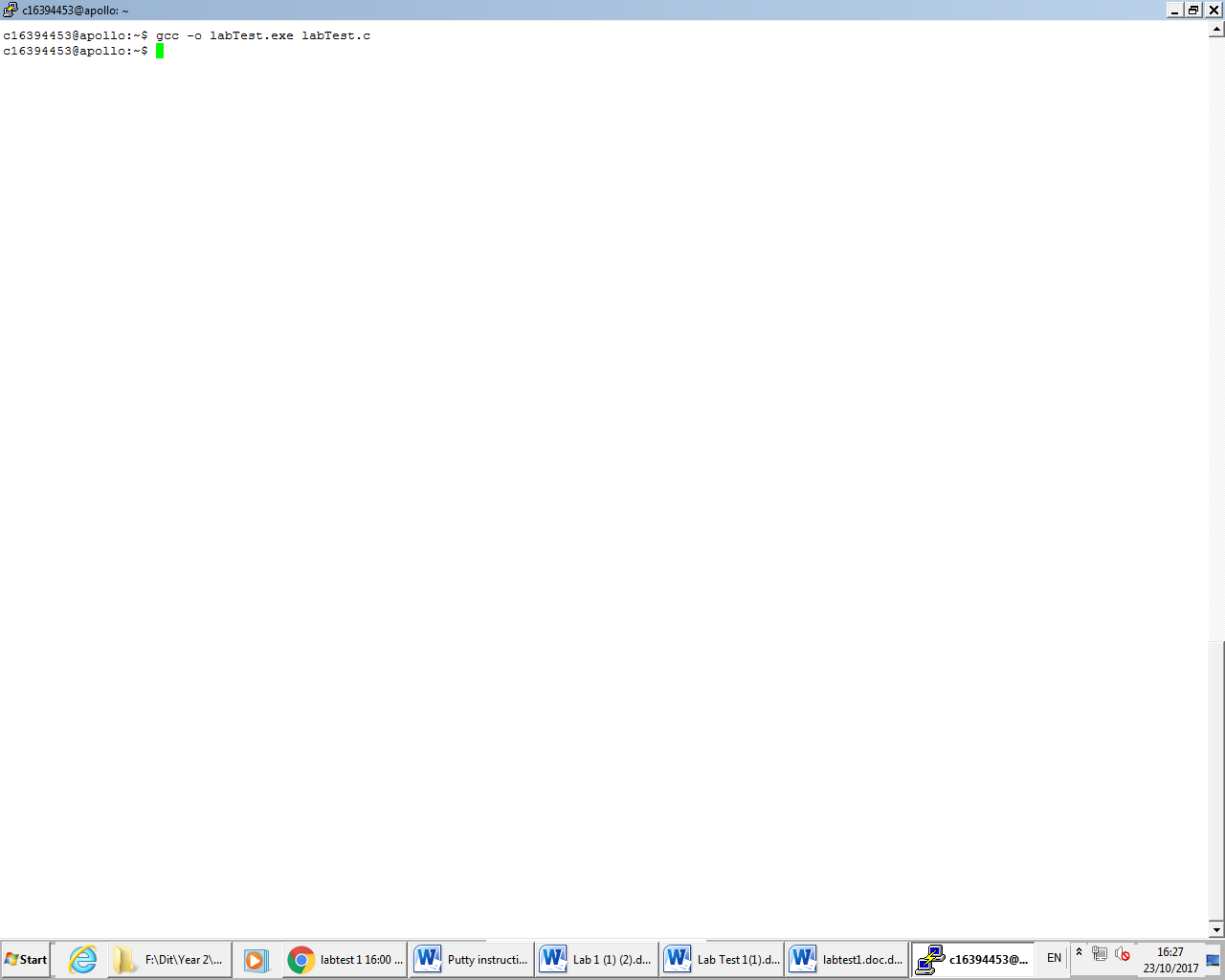
1: The first problem was that ‘headPtr = newPtr;’ was missing a star pointer. This should have been ‘\*headPtr = newPtr;’. The reason is that you need to point to the memory address of headPtr and in this case it is just putting the contents of newPtr into headPtr. We need to pass the value of newPtr to the memory address of headPtr. This wouldn’t of cued up a new node at the head before.

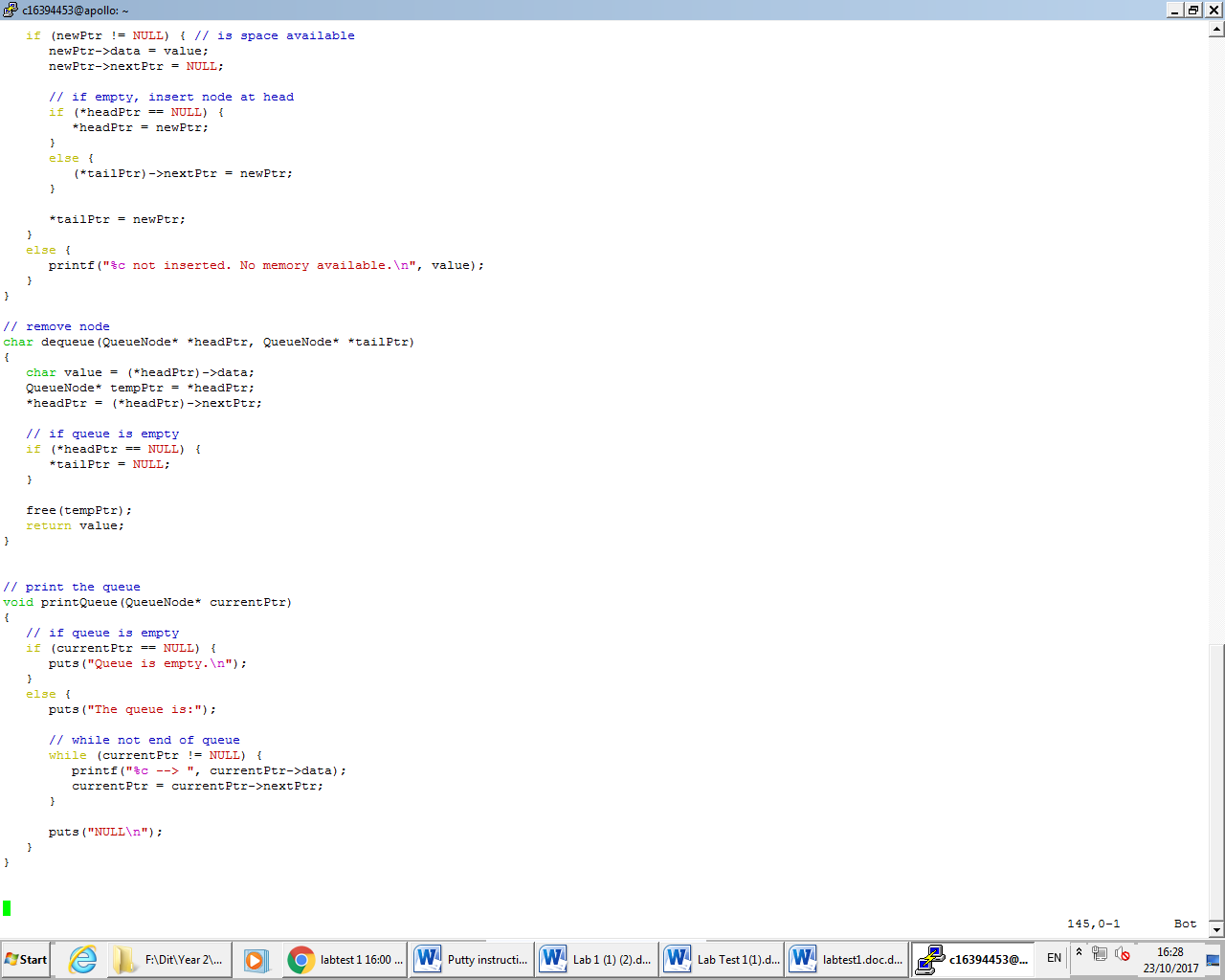


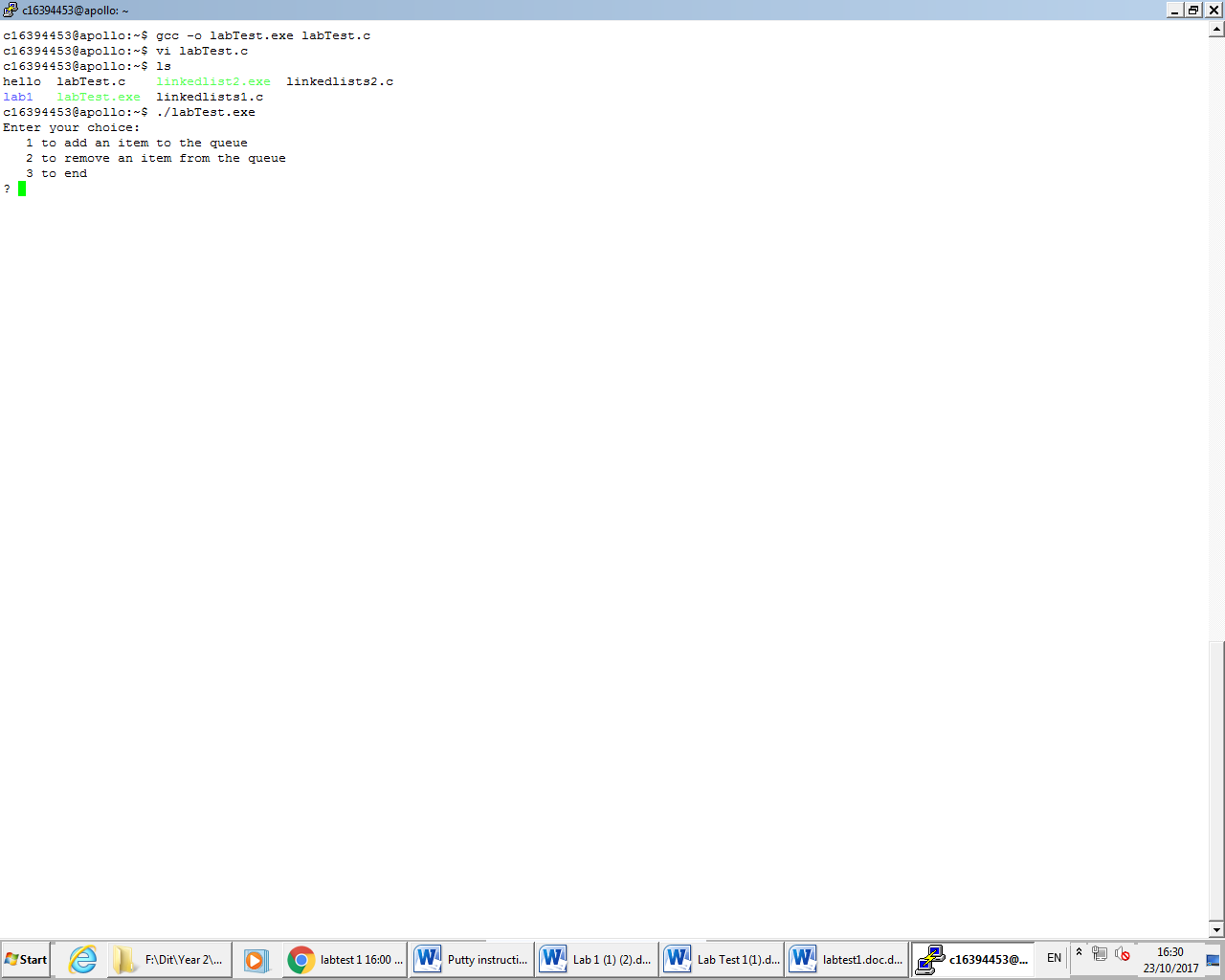


2: This was the same problem as the first. The problem was presenting with ‘tailPtr = newPtr;’ which again isn’t correct. Like last time we don’t want the contents of newPtr going into tailPtr and changing the content. We want the content of newPtr going to the memory address of tailPtr. So in this case the code should look like this: ‘\*tailPtr = newPtr;’. This wouldn’t have allowed to insert a node at the tail.









3: Like the last two times the error has to do with pointer notation. The code looked like ‘headPtr=(\*headPtr)->nextPtr;’. This was wrong as simply what it is doing is getting the memory address of headPtr and pointing it to nextPtr and then the contents going into headPtr, where the content should be going into the memory address of headPtr. The code should look like ‘\*headPtr=(\*headPtr)->nextPtr;’. This wouldn’t have freed the node before.