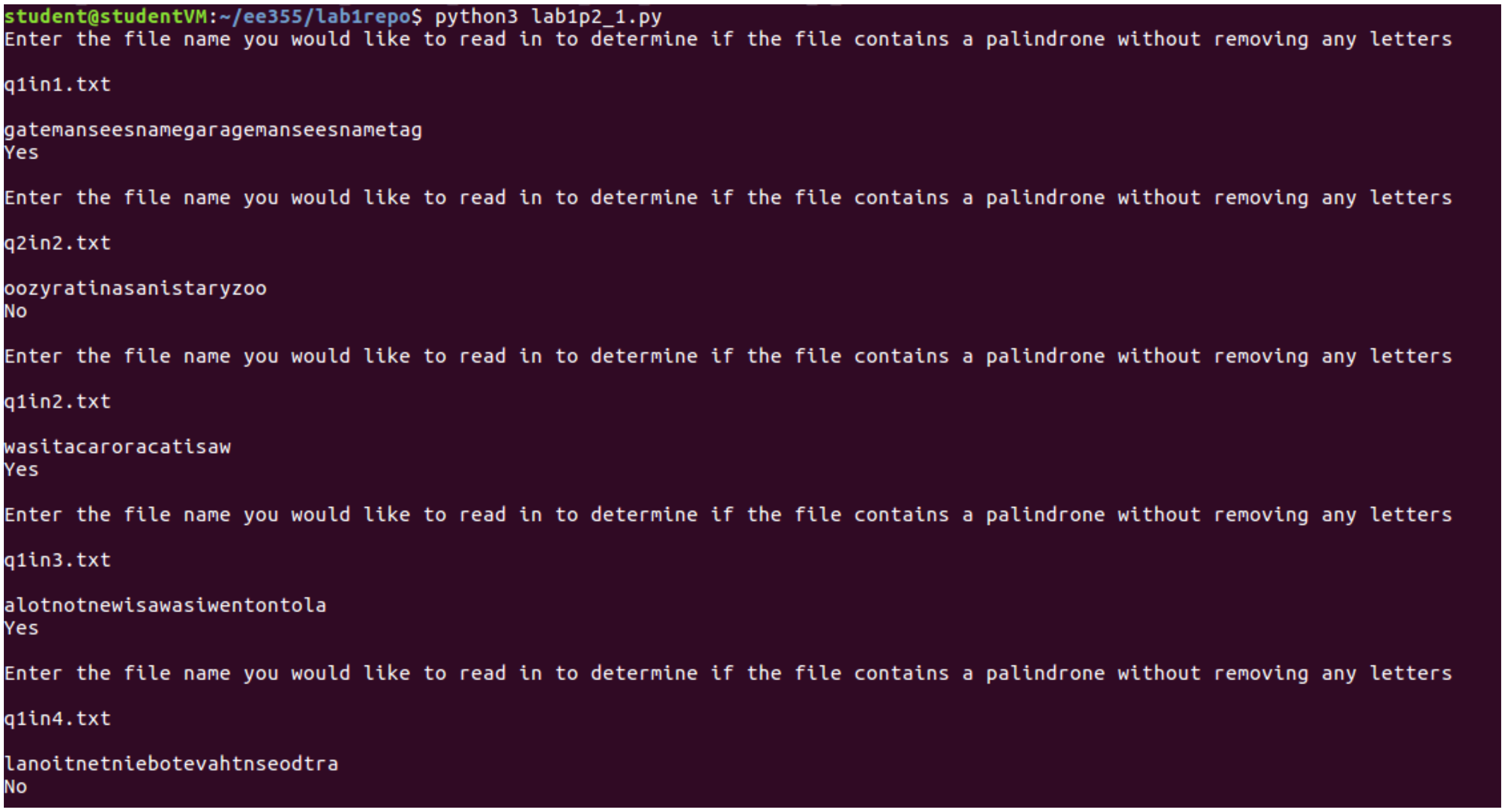
Part 2_1:

For part 2_1 we just had to see if the read-in file contains a palindrome after converting all uppercase letters to lowercase letters and removing all non-alphanumeric characters. First thing I did was put my entire code in a while loop so that after the code completely executes we can just enter another file to read in from. Strings are immutable so I switch them back into lists to remove all casing and non-alphanumerics. After filtering the strings I create a list so that I can easily access each index to then use in a for loop to move all the words at the end of the string to the front, thus creating the comparison string to see if the original string and the comparison string match. If they match, it's a palindrome; otherwise, it is not a palindrome.

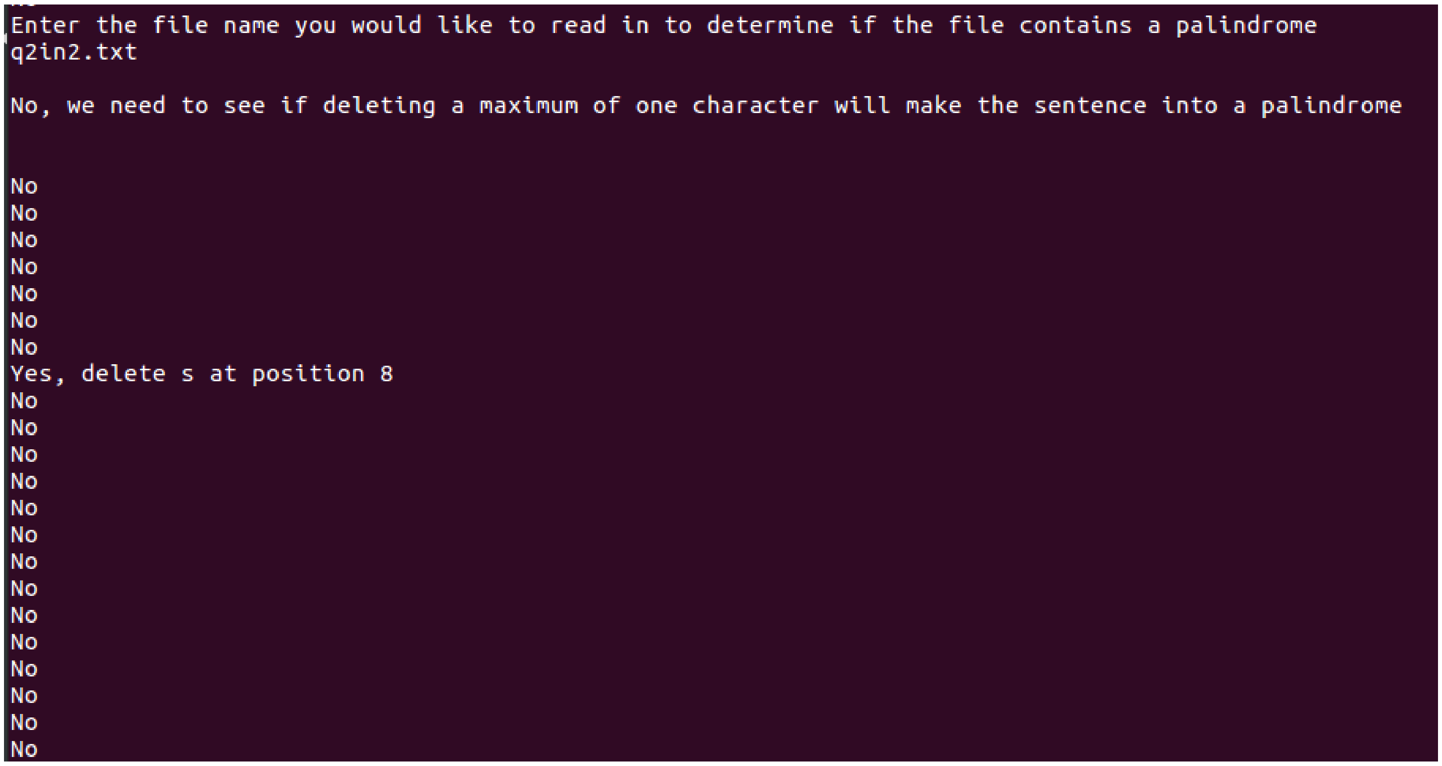


```
student@studentVM:~/ee355/lab1repo$ python3 lab1p2_1.py
Enter the file name you would like to read in to determine if the file contains a palindrone without removing any letters

q1in1.txt

gatemanseesnamegaragemanseesnametag
Yes

Enter the file name you would like to read in to determine if the file contains a palindrone without removing any letters

q2in2.txt

oozyratinasanistaryzoo
No

Enter the file name you would like to read in to determine if the file contains a palindrone without removing any letters

q1in2.txt

wasitacaroracatisaw
Yes

Enter the file name you would like to read in to determine if the file contains a palindrone without removing any letters

q1in3.txt

alotnotnewisawasiwentontola
Yes

Enter the file name you would like to read in to determine if the file contains a palindrone without removing any letters

q1in4.txt

lanoitnetniebotevahtnseodtra
No
```

Part 2_2:

Part 2_2:

For part 2_2 we had to read in a text file and see if it is palindrome after removing all non-alphanumerics and lowercasing everything. If it's not a palindrome then we have to see if we can make it into a palindrome by removing one letter at most. What I did was read in the file then filter the string. After removing everything, I counted the number of words in the original string so that I can have a reference for the loops. Then I run the original string into a for loop to see if it is a palindrome before even removing a letter. If it is not a palindrome then I go and remove a letter. What I do then is have a for loop that removes a letter at a specified index then turns that new string around and checks if it is a palindrome. If it still isn't a palindrome then I put the removed letter back at that index and increment the for loop responsible for keeping track of the index at which a letter will be removed. If by chance the for loop catches the new string to be a palindrome, then it saves the letter and the index and prints out that we found a palindrome.



```
Enter the file name you would like to read in to determine if the file contains a palindrome
q2in2.txt

No, we need to see if deleting a maximum of one character will make the sentence into a palindrome


No
No
No
No
No
No
No
Yes, delete s at position 8
No
No
No
No
No
No
No
No
No
No
No
No
No
No
```