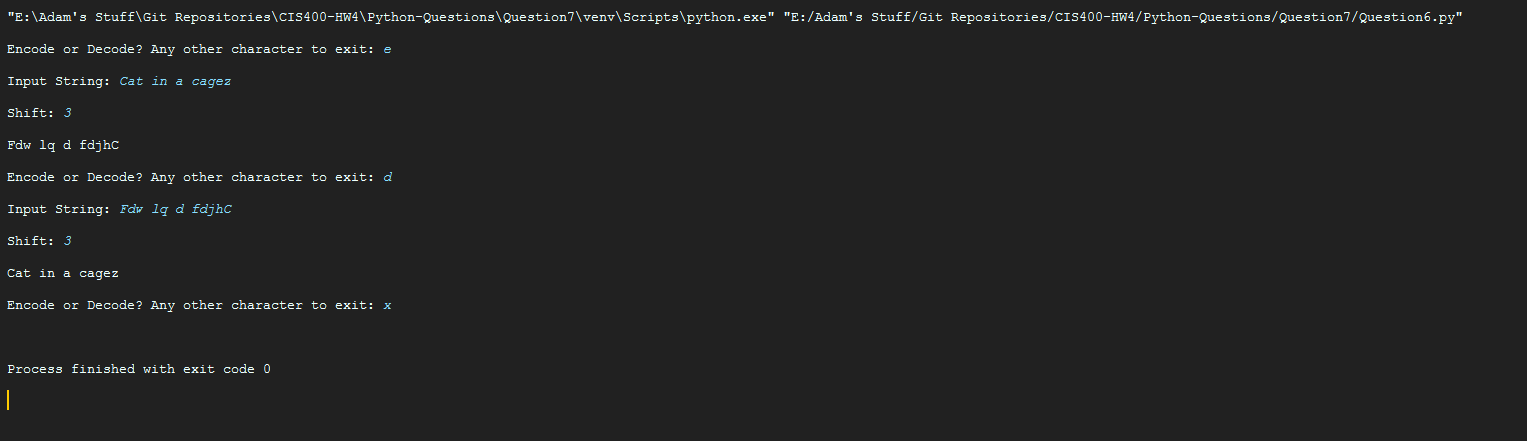
CIS400 Homework 4

Question 6:

-No input checking must be exact,

-Used list of possible characters called alphaList to use as a basis for the resulting encoding or decoding list. Main algorithm includes taking each character from the input string, finding its index in alphaList and then applying the shift based on encode (add) or decode (subtract). The algorithm also handles wrapping by adding or subtracting the length of alphaList to handle roll over.

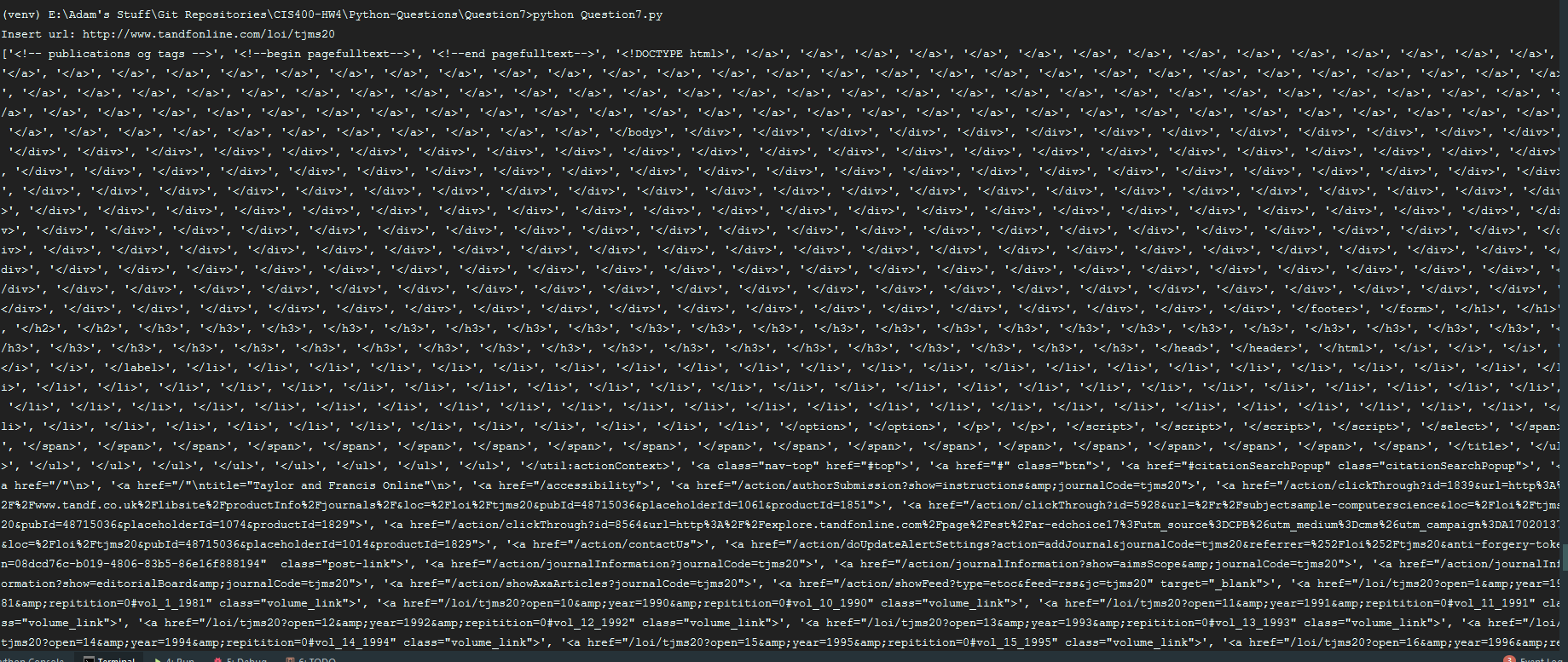
*"""  
 Question 6  
 Python  
 Adam Giaccaglia  
"""  
  
  
def* encode\_or\_decode(type, str, shift):  
 alphaList = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z',  
 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']  
  
 outputLis = []  
 *for* char *in* str:  
 *if* char *is* ' ':  
 outputLis.append(' ')  
 *else*:  
 *if* type *in* "Encode e E".split(" "):  
 pos = alphaList.index(char) + shift  
 *# check and handle wrap  
 if* pos > len(alphaList):  
 pos -= len(alphaList)  
 outputLis.append(alphaList[pos])  
 *if* type *in* "Decode d D".split(" "):  
 pos = alphaList.index(char) - shift  
 *# check and handle wrap  
 if* pos < 0:  
 pos += len(alphaList)  
 outputLis.append(alphaList[pos])  
 *return* outputLis  
  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 e\_or\_d\_input = input("Encode or Decode? Any other character to exit: ")  
 *while* e\_or\_d\_input *in* "Encode e E Decode d D".split(" "):  
 inputStr = input("Input String: ")  
 shift = int(input("Shift: "))  
 result = encode\_or\_decode(e\_or\_d\_input, inputStr, shift)  
 print(''.join(result))  
 e\_or\_d\_input = input("Encode or Decode? Any other character to exit: ")



Question 7

-Algorithm: Use requests module to get html. Use re module and regex expression to grab all the tags and put them in a list. Then, sort the list alphabetically

*"""  
 Question 7  
 Python  
 Adam Giaccaglia  
"""  
  
import* requests  
*import* re  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 inputUrl = input("Insert url: ")  
 res = requests.get(inputUrl)  
 res.raise\_for\_status()  
  
 tagRegex = '<[^>]\*>'  
 findAll = re.findall(tagRegex, res.text)  
 sortedList = sorted(findAll)  
  
 print(sortedList)



Continues but should be enough to show alphabetical sorting

Question 8