CECS 228: Lab Assignment #16

Submission Instructions:

Attach your coded solution to the programming tasks below. When you are finished...

1. Rename this file so that the last names of everyone in your group replaces "YOUR NAME" in the current notebook name, and submit it to the dropbox by **Sunday 12/6 @ 11:59 PM**. For example, I would submit to the dropbox a file called `CECS 228 Coded Assignment #4 - VARELA - DOE.ipynb`
2. Submit **your code only to CodePost as `LAB16.py` by Sunday 12/6 @ 11:59 PM**

Task

Write a function `order(words)` which takes in the *set* of strings `words` and returns a *list* of strings ordered in alphabetical order (lexicographic order). To do so, you must take the following steps:

1. For each word $l_1 l_2 \cdots l_n$ in the set, convert each letter l_i into its unicode representation u_i (look into the built-in function `ord()`).
2. Create an n-tuple for each word using the unicode representations: $l_1 l_2 \cdots l_n \to (u_1, u_2, \ldots, u_n)$.
3. Compare the n-tuples using the lexicographic ordering $(u_1, u_2, \ldots, u_n) \preceq (v_1, v_2, \ldots, v_m)$ if at least one of the following is true:

   - $u_1 < v_1$

   OR

   - $u_i = v_i$ for $i = 1, 2, \ldots, k$ AND $u_{k+1} < v_{k+1}$ where $k + 1 < n, m$

   OR

   - $u_i = v_i$ for $i = 1, 2, \ldots n$ AND $n < m$
4. Order the tuples according to the comparisons carried out in step 3.
5. Convert each tuple back to letters and concatenate to form a word (look into the built-in function `chr()`).
6. Store the result in to a list and return the list.

In [ ]:
```
def order(words):
    """
    orders the words in the given set in alphabetical order
    INPUT: words - set of strings
    OUTPUT: list of strings
    """
    """
    orders the words in the given set in alphabetical order
    INPUT: words - set of strings
    OUTPUT: list of strings
    """
    wordSet = list(words)
    asciiVals = []
    ans = []

    for i in range(len(wordSet)):
        wordSet[i] = wordSet[i].lower()
    for word in wordSet:
        for ch in word:
            asciiVals = [tuple(ord(ch) for ch in word) for word in wordSet]
    while asciiVals:
        minn = asciiVals[0]
        for i in asciiVals:
            if i < minn:
                minn = i
        ans.append(minn)
        asciiVals.remove(minn)
    lastArray = []
    for i in ans:
        charString = ""
        for k in i:
            charString += chr(k)
        lastArray.append(charString)
    return lastArray
```