

Nag a Ram

Purpose: vectors, sort

Due: January 30th

Two words are anagrams of each other if one can be produced by a reordering of letters of the other. For example “resistance” and “ancestries” are a pair of anagrams. Another anagram pair is “admirer” and “married”. Anagrams form an equivalence relation since it is reflexive (each word is an anagram of itself), symmetric (if w_1 is an anagram of w_2 then w_2 is an anagram of w_1) and transitive (if w_1 is an anagram of w_2 and w_2 is an anagram of w_3 then w_1 is an anagram of w_3). Given a list of words you are to separate the words into equivalence classes and list the classes that have the greatest number of members in lexicographical order.

Input: The input will consist of a single test case. The first line contains 2 integers $0 < m, n$ (separated by a single space) which indicates the number of words to consider and the number of classes to print out. On the next n lines will be a single word composed of lowercase alphabetical characters.

Output: Output the n largest anagram equivalence classes. If there are less than n classes, then output them all. Sort the classes by decreasing size. Break ties lexicographically by the lexicographical smallest element. For each class output, print its size and its member words. Sort the member words lexicographically and print equal words only once.

Sample Input

```
16 5
undisplayed
trace
tea
singleton
eta
eat
displayed
crate
cater
carte
caret
beta
beat
bate
ate
abet
```

Sample Output

Class of size 5: caret carte cater crate trace .
 Class of size 4: abet bate beat beta .
 Class of size 4: ate eat eta tea .
 Class of size 1: displayed .
 Class of size 1: singleton .

How the program will be graded

Memo

What	pts
Name	1
Time Analysis $O()$ of every function ^{1,2} (in terms of the words input)	10
Space Analysis $O()$ of every function ^{1,2}	10
Test Plan ³ with at least 4 original nontrivial tests ⁴	10

Source Code Document

What	pts
Name	1
Description ⁵	4
Style	8
pre/post conditions	7
Functionality using the STL	50

¹The main() is a function.

²All analysis should be worst case based on the number of input words.

³A test plan is a table with 4 columns and 1 row per test. The columns are named Reason for the test, actual input data, expected output data, and actual output. You do NOT have to have a working program to write a test plan. Each reason should be unique.

⁴A non trivial test contains only legal data (data that conforms to the input specification) with lists of words that contain at least 1 word .

⁵The description should be written to some one who knows NOTHING about the program. It should discuss what the program does (in your own words). After reading the description the user should be able to create legal input and predict the output.

Required starter code c++, python

```
1 class EquiClasses{
2 private:
3     vector<vector<string>>> classes;
4 public:
5     void add(string word) {
6         /* Adds word to the appropriate class if no anagram class exists
7            creates a new class
8         */
9     }
10    void sortClasses() {
11        /*
12         Sorts the classes by decreasing size
13        */
14    }
15    void printClasses(int num2Print) {
16        /*
17         prints all elements of the first num2Print classes
18        */
19    }
20 };
```

```
1 class EquiClasses:
2     def __init__(self): # constructor
3         self.classes = [[]]
4
5     def add(word) #Adds word to the appropriate class
6                     #if no anagram class exists creates a new class
7     def sortClasses()# Sorts the classes by decreasing size
8
9     def printClasses(num2Print)#Prints all the elements in the first num2Print
    classes
```