Lab 4

Student ID:

4A Problem Description:

A word is "paired-isogram" if each letter in the given word appears exactly twice (not less, not more) in the word. For example, the word teammate is a paired-isogram since each letter in the word appears exactly twice (not less, not more), but the word dad is not since the letter "a" of the word does not appear twice. Write a Java program to determine whether a given word is a paired-isogram.

Input Specification:

Input to the program is given in one line which contain one word with character length not exceeding 52. Input word will contain only lowercase letters (no other characters) and will be at least one letter long.

Output Specification:

The output should be one line where you print all the unique alphabets in sorted order followed by yes if the given word is a paired-isogram. You should print all the non-repeating alphabets in a sorted order followed by no if it the word is not a paired-isogram. All the alphabets prior to yes or no should be separated by a single space.

Hint: ASCII values for [a-z] are 97-122 and [A-Z] are 65-90.

Sample Test Case:

SR. No	Input	Output
1	teammate	a e m t yes
2	dad	a no
3	ali	a i l no
4	dood	d o yes
5	immorior	i m o r yes

Note: **No extra spaces, tabs, blank lines, punctuations, or any other superfluous characters are allowed. You must use only one scanner in your code.**

4B Problem Description:

An “n-gram” is a contiguous sub-sequence of *n* items in a given sequence. For example, given the sequence “ALIGATOR”, its only 5-grams are ALIGA, LIGAT, IGATO and GATOR. There are special names for the first few n-grams: unigram for 1-gram, bigram for 2-gram, and trigram for 3-gram.

Write a program that, given a paragraph, will find the most-frequently appearing n-gram in it. We are interested in n-grams consisting of only alphabets. More specifically, you need to find “the single letter that appears the most (Unigram)”, “two consecutive letters that appear the most (Bigram)”, and “three consecutive letters that appear the most (Trigram)”. If there is a tie for number of occurrences, (*e.g.*, several bigrams appearing the most), you must print one that appears first alphabetically (*i.e.*, the smallest in string comparison). Note that "consecutive" letters mean one letter immediately after another letter, *i.e.*, no other characters (spaces or other separators) in between.

Input Specification:

The first line of the input is an integer p (0<p<51) indicating the number of lines in the paragraph. The following p input lines provide the text for the paragraph. Each of these input lines will contain only lowercase letters, spaces, commas and periods. Assume that these input lines will not exceed column 70 and that each line will contain at least one letter. (Note that the only separators are spaces, commas, periods, and end-of-line.) (p+2)th line or the last line in the input is an integer n (0<n<4). It indicates n-gram to be printed as the output (1->Unigram, 2->Bigram and 3-> Trigram).

Output Specification:

Print appropriate “n-gram” text (Unigram/Bigram/Trigram) followed by most-frequently appearing n-gram. Note that for a string such as “aaaaaa”, some interpretations view it as having three copies of “aa” and some view it as having five occurrences of “aa”. Use the latter view for this problem (same concept applies to trigrams as well).

Sample Test Case:

No	Sample Input	Sample Output
1	2 abcd z z. z,z. z z. z,z. 1	Unigram z
2	3 a a. a,a. bc bc abcd abcd abcd 2	Bigram bc
3	1 abababababababababa 3	Trigram aba