**Intervals**

**Time Remaining**

**1:02**

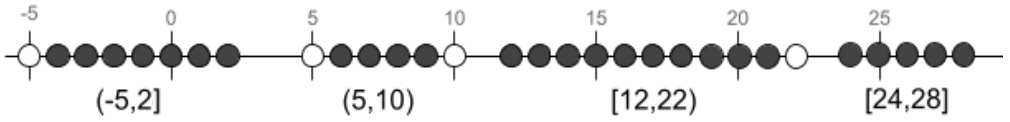
**hrs:mins**

**CLASS / SOURCE NAME**

**Name your class acslintervals**

**PROBLEM:**You will be given three sets of intervals, all representing integers. Each interval is specified as a pair of numbers enclosed in brackets and/or parentheses. Brackets include the end integer and parentheses do not include it. For example, the interval [1,4) represents the integers 1, 2, and 3; the interval (4,6] represents the integers 5 and 6. The set [1,4) and [6,7] are the integers 1, 2, 3, 6, and 7. The intervals in a single set of intervals will not overlap. The first number in each interval will not be greater than the second number.

Each set of intervals is a single string; the only spaces will be a single space between each interval. For example, the string “(5,10) (-5,2] [24,28] [12,22)” contains 4 intervals shown graphically as follows:



If the third string is “null”, then that data set will contain only two sets of intervals. For such data sets, output the sum of the integers that are in one set of intervals, but not the other.

For the data sets with three sets of intervals, output 2 numbers: the sum of the integers in exactly 1 of the 3 sets of intervals, followed by the sum of the integers in exactly 2 of the 3 sets of intervals.

**EXAMPLE - Sample Data #1 with 2 sets of intervals:**

(5,10) (-5,2] [24,28] [12,22)

[1,4] [15,25) [-4,-1) (7,12]

null

The common intervals are: [-4,-2], [1,2], [8,9], [12,12], [15,21], and [24,24].

The integers that are not in these common intervals, but are in at least 1 of the other intervals are:

-1, 0, 3, 4, 6, 7, 10, 11, 13, 14, 22, 23, 25, 26, 27, and 28.

The sum of these integers is 218.

**EXAMPLE - Sample Data #3 with 3 sets of intervals:**

(5,10] [24,25) [13,20] (0,4)

[1,5) [25,26] (8,12] (15,24)

[4,12] [-5,3) (12,25]

The common intervals are: [1,2], [9,10], and [16,20].

The integers that are not in these common intervals, but are in only 1 of the other intervals are:

-5, -4, -3, -2, -1, 0, 5, and 26. The sum of these integers is 16.

The integers that are not in these common intervals, but are in exactly 2 of the other sets of intervals are: 3, 4, 6, 7, 8, 11, 12, 13, 14, 15, 21, 22, 23, 24, and 25. The sum of these integers is 208.

The output for this input is 16 208.

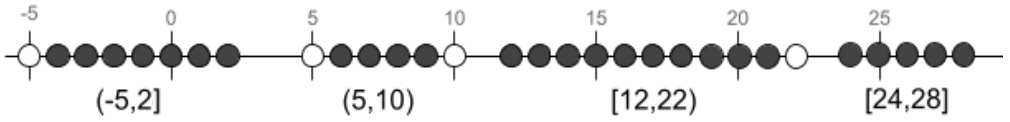
**INPUT:**Input three strings representing the separate lists of non-overlapping intervals of integers. The integers will be between -99 and 99, inclusive. There will be no more than 15 intervals on each line of data.

**OUTPUT:**For each data set, output a string of one integer or two integers separated by a single space. For 2 sets given, output an integer representing the sum of the integers that are in one set of intervals, but not the other, thus not in the common intervals. For 3 sets given, output 2 integers where the first number is the sum of the integers in exactly 1 of the 3 sets of intervals and the second number is the sum of the integers in exactly 2 of the 3 sets of intervals.

|  |  |
| --- | --- |
| **Sample Input** | **Sample Output** |
| (5,10) (-5,2] [24,28] [12,22)  [1,4] [15,25) [-4,-1) (7,12]  null | 218 |
| [0,5) (-15,-5] [8,12] (-4,0)  [3,7) (8,15) [-10,-4) [-4,2] [15,20)  null | 77 |
| (5,10] [24,25) [13,20] (0,4)  [1,5) [25,26] (8,12] (15,24)  [4,12] [-5,3) (12,25] | 16 208 |
| (-84,-80) (-65,-60] [-99,-90) [-80,-70]  (-71,-64) [-95,-91) [-74,-71] (-87,-81]  [-85,-79) (-71,-62] [-97,-87] (-61,-55) | -1313 -1407 |
| (2,6) (13,17) (6,9) (9,13) (0,2)  [15,18] [0,5] [6,14]  (11,14] [3,4) (6,10] [5,5] | 43 69 |

**问题**:给定三组都表示整数的区间，每个区间都表示为一对用方括号和、或圆括号括起来的数字。方括号表示包含区间末端的整数，而圆括号则不包含。例如，区间 [1,4) 表示整数 1、2 和 3；区间 (4,6] 表示整数 5 和 6。区间 [1,4）和 [6,7] 表示整数 1、2、3、6 和 7。一组中的各个区间不会重叠。每个区间的第一个数字不大于第二数字。

每组区间为一个字符串；每个区间之间只有一个空格。例如，字符串“(5,10)(-5,2][24,28][12, 22)” 包含 4 个区间，如下图所示：



如果第三个字符串为 “null”，则该数据集只包含两组区间。对于这样的数据集，输出仅包含于任意一组区间且不包含于另一组区间中的整数之和。

对于包含三组区间的数据集，请输出 2 个数字：分别是仅包含于三组区间中任意一组区间的整数之和，和仅包含于三组区间中任意两组区间的整数之和。

**示例-数据样本#1，其中包含 2 组区间：**

(5,10)(-5,2][24,28][12,22)

[1,4][15,25)[-4,-1)(7,12]

null

公共区间为：[-4,-2]、[1,2]、[8,9]、[12,12]、[15,21] 和 [24,24]。

不包含于这些公共区间中，但至少包含于其他任意一个区间中的整数是：-1、0、3、4、6、7、10、11、13、14、22、23、25、26、27 和 28。

这些整数之和是 218。

**示例-数据样本#3，其中包含 3 组区间:**

(5,10][24,25)[13,20](0,4)

[1,5)[25,26](8,12](15,24)

[4,12][-5,3)(12,25]

公共区间是：[1,2]、[9,10] 和 [16,20]。

不包含于这些公共区间中，但仅包含于其他任意一个区间中的整数有：-5、-4、-3、-2、-1、0、5 和 26。这些整数之和是 16。

不包含于这些公共区间中，但仅包含于其他任意两组区间中的整数有：3、4、6、7、8、11、12、13、14、15、21、22、23、24 和 25。这些整数之和是 208。

对于这个输入，输出为 16 208。

**输入：**输入三个字符串，分别表示三组不重叠的整数区间。整数介于 -99 至 99 之间，包括 -99 和 99。每行数据上的区间不会超过 15 个。

**输出：**对于每个数据集，输出含一个整数的字符串或由一个空格分隔的两个整数的字符串。如果给定两组区间，请输出一个整数，表示包含于任意一组区间且不包含于另一组区间中的整数之和（即不位于公共区间中的整数）。如果给定三组区间，请输出两个整数，其中第一个数是三组区间中仅包含于任意一组区间的整数之和，第二个数是三组区间中仅包含于任意两组区间的整数之和。

|  |  |
| --- | --- |
| **样本输入** | **样本输出** |
| (5,10) (-5,2] [24,28] [12,22)  [1,4] [15,25) [-4,-1) (7,12]  null | 218 |
| [0,5) (-15,-5] [8,12] (-4,0)  [3,7) (8,15) [-10,-4) [-4,2] [15,20)  null | 77 |
| (5,10] [24,25) [13,20] (0,4)  [1,5) [25,26] (8,12] (15,24)  [4,12] [-5,3) (12,25] | 16 208 |
| (-84,-80) (-65,-60] [-99,-90) [-80,-70]  (-71,-64) [-95,-91) [-74,-71] (-87,-81]  [-85,-79) (-71,-62] [-97,-87] (-61,-55) | -1313 -1407 |
| (2,6) (13,17) (6,9) (9,13) (0,2)  [15,18] [0,5] [6,14]  (11,14] [3,4) (6,10] [5,5] | 43 69 |

注意：

(1) 样本数据仅为部分测试数据，测试用例全部通过不代表通过本题。

(2) 你必须通过数据库中所有的测试点才能获得该题满分。

(3) java 语言里面的 class name（类名）需要用本题的 Source file name（即：acslintervals）。

(4) 平台判分规则为调取 10 次 input 值，每次单独判分。

**KWIC**

**Time Remaining**

**1:01**

**hrs:mins**

**CLASS / SOURCE NAME**

**Name your class acslkwic**

**PROBLEM:**A Key Word in Context (KWIC) is an index of words in a document, along with the context of each word. Unimportant words (e.g, the, will, have, that, it, and, after, is) are not indexed.

In this problem, you are given three inputs: a string to be indexed according to each of its words, a string of unimportant words, and a string for a range of integers. A single space will separate all items in each string. Find all of the words in the first string that are not included in the second string of unimportant words. Create an alphabetized table of these words using the following rules:

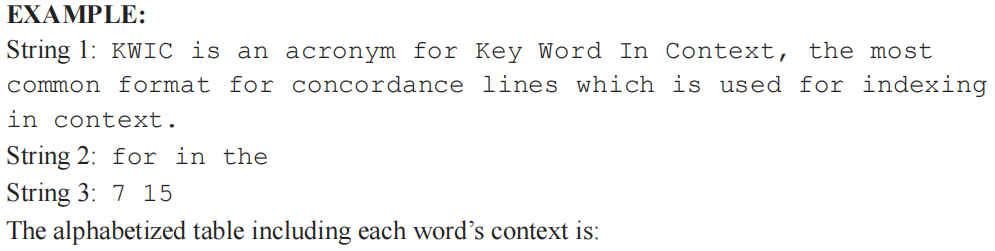
* The unimportant words are given as lowercase, but they are not included in the final table regardless of their case in the string
* Every other word in the first string will be included in the table exactly as it is found in the string
* If a word occurs more than once in the first string, add each occurrence to the table in the order that the word appears in the string
* Alphabetize the words in the table regardless of case without rearranging identical words

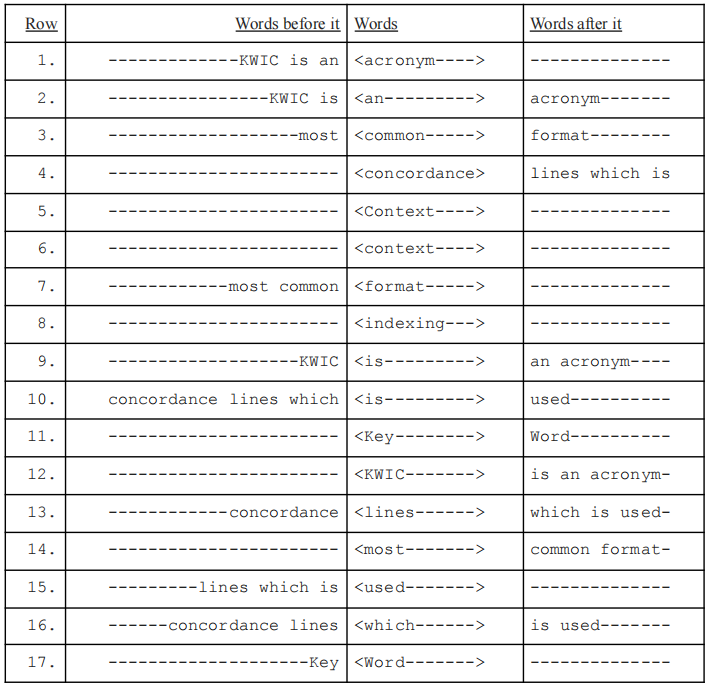
The table will have three columns that also include each word’s context. The first column contains the words before it; the second column is the word itself; and the third column contains the words after it. Use the following rules for the left side and then the right side of the indexed word:

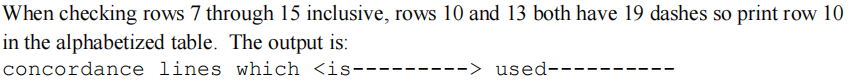
* Find no more than 3 words before or after the indexed word
* Stop if you find a punctuation mark (i.e. period, question mark, exclamation point, comma, semi-colon, colon)
* Stop if you find any of the unimportant words

Dashes are added to the table to right-justify the first column and left-justify the other two columns. Include a single space between each column. Surround the indexed word with a “<” and “>” symbol.

The table starts with Row 1. The third input will identify the range of row numbers, including both numbers, that must be searched in the alphabetized table. Within the inputted range of rows in the table, find the row that has the fewest number of dashes. If there is a tie, use the first one found. Output that entire row in the table.







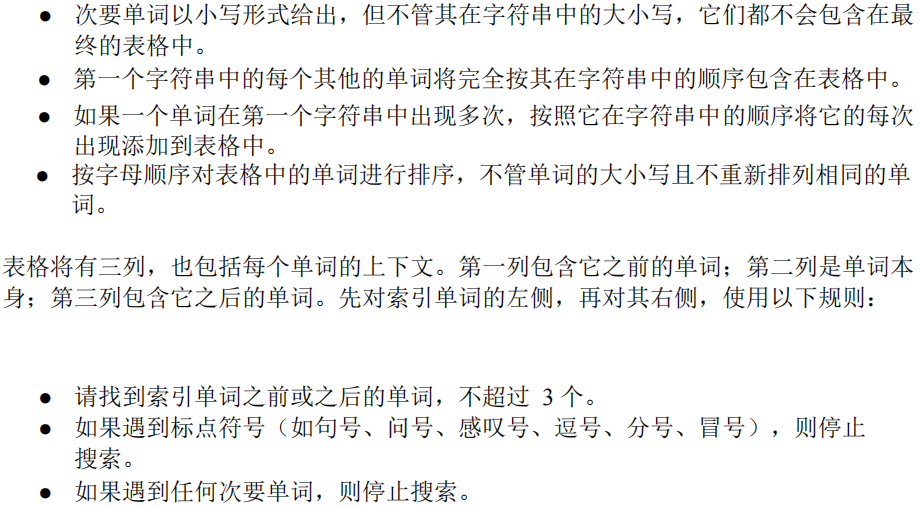
**INPUT:**Each set of data will contain 3 strings, the string to be indexed, the string of unimportant words, and a string of two integers separated by a single space identifying the inclusive range of rows to be checked in the alphabetized table. All items in each string will be separated by a single space. The first string will be no longer than 500 characters and will be case sensitive as shown above. Only spaces, punctuation, and alphabetic characters will be found in the string. All unimportant words will be in lowercase. The integers will be a valid inclusive range of row numbers in the table, both greater than or equal to 1.

**OUTPUT:**Output the row in the table, within the given range, that has the fewest number of dashes. If there is a tie, print the one that comes first in the alphabetized table. Use the correct spacing so that the alphabetized words are lined up as shown in the table above. Use a “-” to represent any spaces needed to right-justify the first column and left-justify the other 2 columns. Your answer must use exactly the same formatting as described above.

|  |  |
| --- | --- |
| **Sample Input** | **Sample Output** |
| KWIC is an acronym for Key Word In Context, the  most common format for concordance lines which is  used for indexing in context.  for in the  7 15 | concordance lines which <is---------> used---------- |
| The quick, brown fox jumped over the lazy dogs.  the  1 6 | ---------------- <brown-> fox jumped over |
| HackerRank is the best platform to do ACSL programs  for the Finals competition.  for the  1 10 | best platform to <do---------> ACSL programs--- |
| How are you? I am fine. It is nice to be in Florida.  Writing Python programs is easier by the pool!  to in by the  5 15 | Writing Python programs <is------> easier------------ |
| Lions and Tigers and Bears, Oh My! is from the Wizard  of Oz. Where is it found?  and from the it  1 10 | Wizard of <Oz----> ----- |

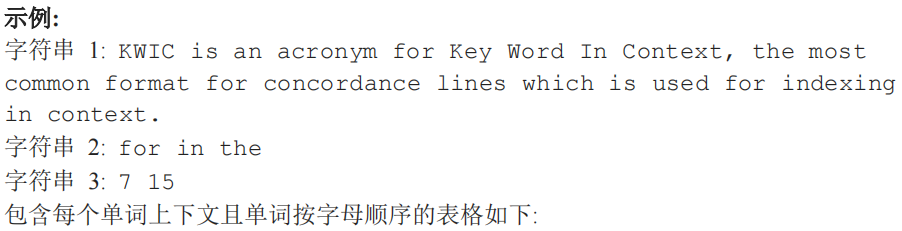
**问题：**上下文关键词索引（KWIC）是对文档中单词的索引，且包含其上下文。次要单词（如 the、will、have、that、it、and、after、is）不会被索引。

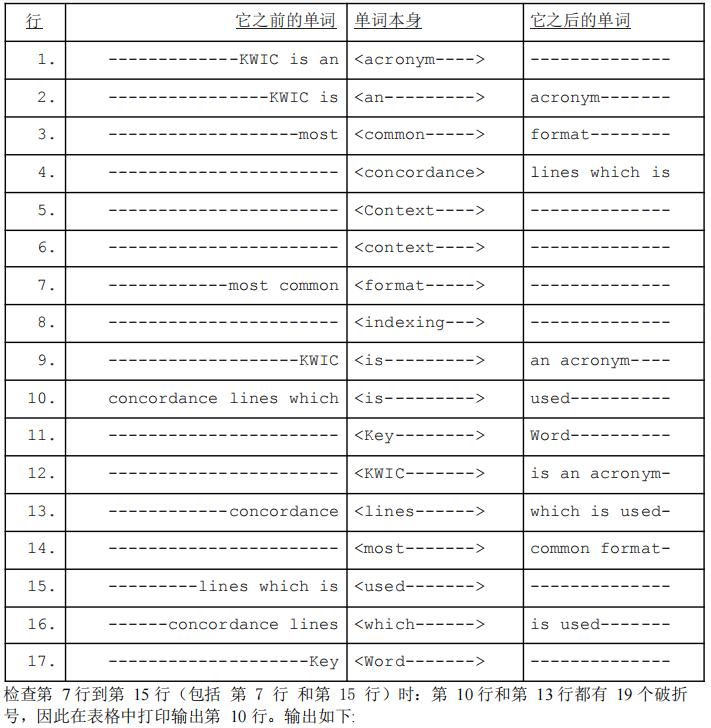
在这个问题中，你将会获得三个输入：一个要根据其每个单词进行索引的字符串，一个包含次要单词的字符串，以及一个表示整数范围的字符串。每个字符串中的各项之间以一个空格分隔。找出包含于第一个字符串中但不包含于第二个次要单词字符串中的所有单词。根据以下规则创建这些单词按字母顺序排列的表格：



在表格中，使用破折号将第一列右对齐，将其他两列左对齐。每列之间用一个空格分隔。用 “<” 和 “>” 符号将索引单词括起来。

表格从第一行开始。第三个输入包括两个数字，将标识出必须在按字母顺序的表格中搜索到的行号范围。在表格中已输入行号的范围内，找到破折号最少的行。如果存在多个最少的行，请使用找到的第一个。请在表格中输出这一整行。





concordance lines which <is---------> used----------

**输入:**每组数据将包含 3 个字符串，进行索引的字符串，次要单词的字符串，以及由单个空格分隔的两个整数字符串，用于标识出在按字母顺序的表格中要检查的行号范围。每个字符串中的各项将用一个空格分隔。第一个字符串的长度不超过 500个字符，并且将严格如上述所示区分大小写。字符串中只包含空格、标点符号和字母。所有的次要单词都将以小写形式给出。两个整数将是表格中的有效行号范围（包含开始和结束的行号），两个整数都大于或等于 1。

**输出:**输出按字母顺序的表格中第一个最少破折号的行。如果存在多个最小值，打印输出按字母顺序的表格中的第一个行。请使用正确的空格间距，使按字母顺序排序的单词在上述表格中对齐。请使用“-”表示所需的任何空格，使第一列右对齐，其他两列左对齐。你的答案必须与上述格式完全一致。

|  |  |
| --- | --- |
| **样本输入** | **样本输出** |
| KWIC is an acronym for Key Word In Context, the  most common format for concordance lines which is  used for indexing in context.  for in the  7 15 | concordance lines which <is---------> used---------- |
| The quick, brown fox jumped over the lazy dogs.  the  1 6 | ---------------- <brown-> fox jumped over |
| HackerRank is the best platform to do ACSL programs  for the Finals competition.  for the  1 10 | best platform to <do---------> ACSL programs--- |
| How are you? I am fine. It is nice to be in Florida.  Writing Python programs is easier by the pool!  to in by the  5 15 | Writing Python programs <is------> easier------------ |
| Lions and Tigers and Bears, Oh My! is from the Wizard  of Oz. Where is it found?  and from the it  1 10 | Wizard of <Oz----> ----- |

注意：

(1) 样本数据仅为部分测试数据，测试用例全部通过不代表通过本题。

(2) 你必须通过数据库中所有的测试点才能获得该题满分。

(3) java 语言里面的 class name（类名）需要用本题的 Source file name（即：acslkwic）。

(4) 平台判分规则为调取 10 次 input 值，每次单独判分。