819. Word Sorting

* [Description](http://www.lintcode.com/en/problem/word-sorting/" \l "description)
* [Notes](http://www.lintcode.com/en/problem/word-sorting/#note)
* [Testcase](http://www.lintcode.com/en/problem/word-sorting/#testcase)
* [Judge](http://www.lintcode.com/en/problem/word-sorting/#judge)

Give a new alphabet, such as {c,b,a,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}. Sort the string array according to the new alphabet

 Notice

* The length of word does not exceed 100.
* The number of words does not exceed 10000.
* You can assume that the given new alphabet is a 26-character string.
* Only lowercase letters.

Have you met this question in a real interview?

Yes

**Example**

Given Alphabet = {c,b,a,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}, String array = {cab,cba,abc}, return {cba,cab,abc}.

Explanation:

According to the new dictionary order, output the sorted result {cba, cab, abc}.

Given Alphabet = {z,b,a,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,c}, String array = {bca,czb,za,zba,ade}, return {zba,za,bca,ade,czb}.

Explanation:

According to the new dictionary order, output the sorted result {zba,za,bca,ade,czb}.

<http://www.lintcode.com/en/problem/word-sorting/#>

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication16;

**import** java.util.Arrays;

**import** java.util.HashMap;

***/\*\****

***\****

***\* @author Usuario***

***\*/***

**public** **class** JavaApplication16 {

***/\*\****

***\* @param args the command line arguments***

***\*/***

**static** **int** partition(String arr[], String[] paralelo, **int** low, **int** high)

    {

        String pivot = paralelo[high];

**int** i = (low-1); *// index of smaller element*

**for** (**int** j=low; j<high; j++)

        {

*// If current element is smaller than or*

*// equal to pivot*

**if** (paralelo[j].compareTo(pivot)  < 0)

            {

                i++;

*// swap arr[i] and arr[j]*

                String temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

                String tp = paralelo[i];

                paralelo[i] =paralelo[j];

                paralelo[j] = tp;

            }

        }

*// swap arr[i+1] and arr[high] (or pivot)*

        String temp = arr[i+1];

        arr[i+1] = arr[high];

        arr[high] = temp;

         String tp = paralelo[i+1];

         paralelo[i+1] =paralelo[high];

         paralelo[high] = tp;

**return** i+1;

    }

*/\* The main function that implements QuickSort()*

*arr[] --> Array to be sorted,*

*low  --> Starting index,*

*high  --> Ending index \*/*

**static** **void** quick(String arr[], String[] paralelo, **int** low, **int** high)

    {

**if** (low < high)

        {

*/\* pi is partitioning index, arr[pi] is*

*now at right place \*/*

**int** pi = partition(arr,paralelo, low, high);

*// Recursively sort elements before*

*// partition and after partition*

            quick(arr,paralelo, low, pi-1);

            quick(arr, paralelo, pi+1, high);

        }

    }

**public** **static** String[] wordSort(**char**[] alphabet, String[] words) {

*// Write your code here*

*//char[] Alphabet = {'c','b','a','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};*

*//String array = {cab,cba,abc};*

*//return {cba,cab,abc}.*

        HashMap<Character, Character> hm =

**new** HashMap();

**int** i =0;

**for**(**char** ch = 'a'; ch <= 'z'; ch++) {

           hm.put( alphabet[i++], ch);

        }

        String[] arr= **new** String[words.length];

**for**(i = 0; i < words.length; i++) {

            String actual = words[i];

            String concat ="";

**for**(**int** j =0; j<actual.length(); j++) {

                concat += hm.get(actual.charAt(j));

            }

            arr[i] = concat;

        }

        quick(words, arr,0, arr.length-1 );

**return** words;

    }

**public** **static** **void** main(String[] args) {

*// TODO code application logic here*

*/\**

*char[] alpha = {'c','b','a','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};*

*String[] w ={"bca","czb","za","zba","ade"};*

*wordSort(alpha, w );\*/*

*/\**

*char[] alp = {'c','b','a','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};*

*String[] array = {"cab","cba","abc"};*

*\*/*

        String alp = "zbadefghijklmnopqrstuvwxyc";

       String[] array = {"bca","czb","za","zba","ade"};

*//["zba","za","bca","ade","czb"]*

       String[] res = wordSort(alp.toCharArray(), array);

**for**(**int** i =0; i<res.length; i++) {

            System.out.print(res[i] + " ");

       }

    }

}