Final version of Google

import java.util.\*;  
  
public class Level1 {  
  
 public static int [] WordSearch(int len, String s, String subs){  
 String nextStr = *substringExtract*(s, len);  
 String[] splitStr = nextStr.split("\\s+");  
 ArrayList<String> strList5 = new ArrayList<>();  
 for (int i = 0; i < splitStr.length; i++) {  
 strList5.add(splitStr[i]);  
 }  
 int finalArrayLength = *substringMatch*(*strListOfGivenLength*(strList5, len),subs).length;  
 int finalArray[] = new int[finalArrayLength];  
 finalArray = *substringMatch*(*strListOfGivenLength*(strList5, len),subs);  
  
 return finalArray;  
 }  
  
 public static String splitStringByLength(String givenStr, int wordLength1) {  
 int L = (givenStr.length() / wordLength1); //number of iterations (splits) through the string  
 StringBuilder str = new StringBuilder(givenStr);  
 if (givenStr.length() % wordLength1 != 0) {  
 for (int i = 1; i <= L; i++) {  
 str.insert(((wordLength1 \* i + i) - 1), ' '); // inserting ' ' after each L \* i + i position  
 }  
 }  
 else {  
 for (int i = 1; i <= L - 1; i++) {  
 str.insert(((wordLength1 \* i + i) - 1), ' '); // same, but to avoid ' ' at the end of last word  
 }  
 }  
 String splitResult = str.toString();  
 return splitResult;  
 }  
  
 public static String substringExtract (String inputStr, int len1) {  
 String finalString = "";  
 String tempString1 = "";  
 int offset = 0;  
  
 for(int i = 0; i < inputStr.length(); i++){  
  
 if(Character.*isWhitespace*(inputStr.charAt(i))){  
 if (i - offset < len1){  
 finalString = finalString + inputStr.substring(offset, i) + " ";  
 offset = i + 1;  
 }  
 else {  
 tempString1 = *splitStringByLength*(inputStr.substring(offset, i), len1);  
 finalString = finalString + tempString1 + " ";  
 offset = i + 1;  
 }  
 }  
  
 if (i == inputStr.lastIndexOf(' ')){  
 if (inputStr.substring(offset).length() < len1){  
 finalString = finalString + inputStr.substring(offset);  
 }  
 else {  
 tempString1 = *splitStringByLength*(inputStr.substring(offset), len1);  
 finalString = finalString + tempString1;  
 }  
 }  
 }  
 return finalString;  
 }  
  
 public static ArrayList<String> strListOfGivenLength (ArrayList<String> someList, int sLength){  
 int offset = 0;  
 String tempStr = "";  
 ArrayList<String> someArrayList = new ArrayList<>();  
  
 for (int m = 0; m < someList.size(); m++){  
  
 if (offset == 0 && someList.get(m).length() == (sLength - 1) || offset == 0 && someList.get(m).length() == sLength){  
 someArrayList.add(someList.get(m));  
 continue;  
 }  
  
 if (offset + someList.get(m).length() > sLength){  
 someArrayList.add(tempStr);  
  
 //checking if this is the last word in the string  
 if ((m == someList.size() - 1)){  
 someArrayList.add(someList.get(m));  
 break;  
 }  
  
 //checking length against sLength one more time  
 if (someList.get(m).length() >= (sLength - 1)){  
 someArrayList.add(someList.get(m));  
 tempStr = "";  
 offset = 0;  
 continue;  
 }  
 else {  
 offset = someList.get(m).length() + 1;  
 tempStr = someList.get(m);  
 continue;  
 }  
 }  
  
 if (offset + someList.get(m).length() < (sLength - 1)){  
 tempStr = (tempStr + " " + someList.get(m)).trim();  
 offset = offset + someList.get(m).length() + 1;  
 if (m == someList.size() - 1){  
 someArrayList.add(tempStr);  
 break;  
 }  
 continue;  
 }  
  
 if (offset + someList.get(m).length() == (sLength - 1) || offset + someList.get(m).length() == sLength){  
 tempStr = (tempStr + " " + someList.get(m)).trim();  
 someArrayList.add(tempStr);  
 offset = 0;  
 tempStr = "";  
 }  
 }  
 someArrayList.removeAll(Arrays.*asList*("", null));  
 return someArrayList;  
 }  
  
 public static int[] substringMatch(ArrayList<String> someList, String givenString){  
 //returning array of 0s and 1s, where 1 is for list element containing givenString  
 int finalArray[] = new int[someList.size()];  
  
 for (int k = 0; k < someList.size(); k++){  
  
 if (someList.get(k).equals(givenString) || someList.get(k).contains(" " + givenString + " ")){  
 finalArray[k] = 1;  
 }  
 else if (someList.get(k).startsWith(givenString + " ") || someList.get(k).endsWith(" " + givenString)){  
 finalArray[k] = 1;  
 }  
 else {  
 finalArray[k] = 0;  
 }  
 }  
 return finalArray;  
 }  
  
 public static void main(String[] args) {  
 String sss = "1 22 33 44 seven555 666 777 8888 seven 123456 letters1234567890 1";  
 int lll = 7;  
 String searchString = "seven";  
 System.*out*.println(*WordSearch*(lll, sss, searchString));  
 System.*out*.println(Arrays.*toString*(*WordSearch*(lll, sss, searchString)));  
  
  
  
  
  
  
  
  
 }  
  
}