Daily Practice Problem Date 17/11/2023

Problem 1: Substring-with-concatenation-of-all-words

```
package ishwarchavan.com;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;

public class SubstringCancatinationOfAllWord{       //class created

      public static ArrayList<Integer> findSubstring(String A, final List<String> B) {
//function created

            int size_word = B.get(0).length();   // Number of a characters of a word in
list L.

            int word_count = B.size();      // Number of words present inside list L.

            int size_l = size_word * word_count;    // Total characters present in list
L.

            ArrayList<Integer> res = new ArrayList<Integer>();     // Resultant vector
which stores indices.
            int n = A.length();

            if (size_l > n)     //condition checking
            {
                  return res;
            }

            HashMap<String, Integer> hashMap = new HashMap<String, Integer>();
                                                                  // Map stores the
words present in list L
                                                                  // against it's
occurrences inside list L

            for (String word : B)
            {
                  hashMap.put(word, hashMap.getOrDefault(word, 0) + 1);
            }


            for (int i = 0; i <= n - size_l; i++)
            {
                  HashMap<String, Integer> tempMap = (HashMap<String, Integer>)
hashMap.clone();
                  int j = i, count = word_count;

                  while (j < i + size_l)    // Traverse the substring
                  {

                  String word = A.substring(j, j + size_word);    // Extract the word

                        if (!hashMap.containsKey(word) || tempMap.get(word) == 0)
//condition checking
                        {
                              break;
                        }

                                                            // Else decrement the
count of word from hash_map
                        else
                        {
                              tempMap.put(word, tempMap.get(word) - 1);
```

```
                        count--;
                    }
                    j += size_word;
                }

                if (count == 0)
                {
                    res.add(i);
                }

            }
            return res;
        }

        public static void main(String[] args) {  //main program
            String S = "barfoothefoobarman";     //string variable created
            ArrayList<String> L = new ArrayList<>(Arrays.asList("foo", "bar"));
//object created
            ArrayList<Integer> indices = findSubstring(S, L);
            for (Integer i : indices)
            {
                System.out.println(i);
            }
        }
}
```

Problem 2: Insert Position

```
package ishwarchavan.com;
import java.io.*;

public class InsertPosition {    //class created

static int find_index(int[] arr, int n, int K) {    //Function to find insert position
of K

    for(int i = 0; i < n; i++)  // Traverse the array

        if (arr[i] == K)    // If K is found
            return i;

        else if (arr[i] > K)
            return i;

    return n;          // If all elements are smaller than K
}
public static void main(String[] args) {    //main program
    int[] arr = { 1, 3, 5, 6 };
    int n = arr.length;
    int K = 2;

    System.out.println(find_index(arr, n, K));   //calling function
}
}
```