=LAB ASSIGNMENT JAVA=

NAME: SUMIT

UID:12284

Problem:-1

```
public class Problem1 {
    public static String matchFound(String input1, String input2)
{
        String[] words = input2.split(":");
        StringBuilder output1 = new StringBuilder();
        for (String word : words) {
            if (word.length() != input1.length()) continue;
            boolean match = true;
            for (int i = 0; i < word.length(); i++) {
                if (input1.charAt(i) != '_' && input1.charAt(i) !=
word.charAt(i)) {
                    match = false;
                    break;
                }
            }
            if (match) {
                if (output1.length() > 0) output1.append(":");
                output1.append(word.toUpperCase());
            }
        }
        return output1.toString();
    }
```

}

Problem:-2

```
public class Problem2 {
    public static String[] encodeStrings(String s1, String s2,
String s3) {
        String[] p1 = splitThreeParts(s1);
        String[] p2 = splitThreeParts(s2);
        String[] p3 = splitThreeParts(s3);

        String output1 = p1[0] + p2[1] + p3[2];
        String output2 = p1[1] + p2[2] + p3[0];
        String output3 = toggleCase(p1[2] + p2[0] + p3[1]);

        return new String[]{output1, output2, output3};
    }

    private static String[] splitThreeParts(String s) {
        int len = s.length();
        int rem = len % 3;
        int part = len / 3;
        int front = part, mid = part, end = part;
        if (rem == 1) mid++;
```

```
        if (rem == 2) { front++; end++; }

        return new String[]{
            s.substring(0, front),
            s.substring(front, front + mid),
            s.substring(front + mid)
        };
    }

    private static String toggleCase(String str) {
        StringBuilder sb = new StringBuilder();
        for (char ch : str.toCharArray()) {
            sb.append(Character.isUpperCase(ch) ?
Character.toLowerCase(ch) : Character.toUpperCase(ch));
        }
        return sb.toString();
    }
}
```

Problem:-3

```
public class Problem3 {
    public static String transformString(String input) {
        StringBuilder sb = new StringBuilder();
        input = input.toLowerCase();
        int i = 0;
        while (i < input.length()) {
            char c1 = input.charAt(i);
            sb.append(c1);
```

```
            if (i + 1 < input.length()) {
                char c2 = input.charAt(i + 1);
                if (Character.isLetter(c1) && Character.isLetter(c2)) {
                    int val = (c1 - 'a' + 1) + (c2 - 'a' + 1);
                    char toAppend = (val % 26 == 0) ? '0' : (char) ((val
% 26 - 1) + 'a');
                    sb.append(toAppend);
                }
            }
            i++;
        }
        return sb.toString();
    }
}
```

Problem:-4

```
public class Problem4 {
    public static char findTheDifference(String s, String t) {
        int res = 0;
        for (char c : s.toCharArray()) res ^= c;
        for (char c : t.toCharArray()) res ^= c;
        return (char) res;
    }
}
```

Problem:-5

```
import java.util.*;
```

```
public class Problem5 {
    public static int[] nextGreaterElement(int[] nums1,
int[] nums2) {
        Map<Integer, Integer> map = new HashMap<>();
        Stack<Integer> stack = new Stack<>();
        for (int num : nums2) {
            while (!stack.isEmpty() && num > stack.peek())
{
                map.put(stack.pop(), num);
            }
            stack.push(num);
        }

        int[] ans = new int[nums1.length];
        for (int i = 0; i < nums1.length; i++) {
            ans[i] = map.getOrDefault(nums1[i], -1);
        }
        return ans;
    }
}
```