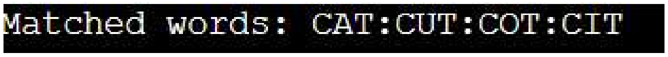
Fast Learner Assignment

Student Name: Sumantra Rijal UID:22BCS50206
Branch: BE. CSE Section/Group: 22BCS_IOT-642 / B
Semester: 6 Code: 22CSP-359
Subject Name: Project based learning in Java Subject

Problem 1.
CODE:

```
public class MatchFinder {
    public 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 isMatch = true;
            for (int i = 0; i < input1.length(); i++) {
                char c1 = input1.charAt(i);
                char c2 = word.charAt(i);
                if (c1 != '_' && c1 != c2) {
                    isMatch = false;
                    break;
                }
            }
            if (isMatch) {
                if (output1.length() > 0) output1.append(":");
                output1.append(word.toUpperCase());
            }
        }
        return output1.toString();
    }
    public static void main(String[] args) {
        MatchFinder finder = new MatchFinder();
        // Example Inputs
        String input1 = "c_t";
        String input2 = "cat:cut:cot:cit:bat";
        // Calling the method and displaying output
        String result = finder.matchFound(input1, input2);
        System.out.println("Matched words: " + result);
    }
}
```

Output:



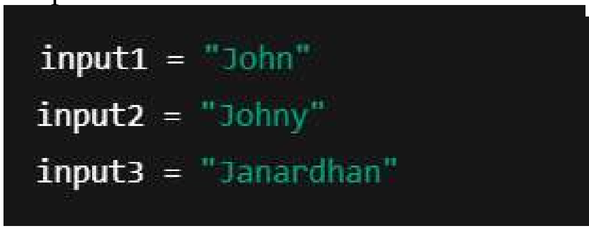
```
Matched words: CAT:CUT:COT:CIT
```

Problem 2.

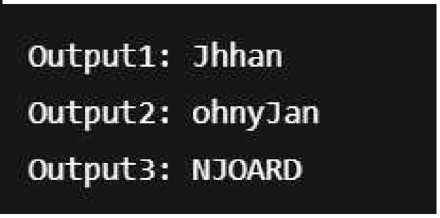
CODE:

```
public class StringEncoder {
    public static String[] splitString(String input) {
        int len = input.length();
        int rem = len % 3;
        int part = len / 3;
        int frontLen = part, middleLen = part, endLen = part;
        if (rem == 1) {
            middleLen += 1;
        } else if (rem == 2) {
            frontLen += 1;
            endLen += 1;
        }
        String front = input.substring(0, frontLen);
        String middle = input.substring(frontLen, frontLen + middleLen);
        String end = input.substring(frontLen + middleLen);
        return new String[]{front, middle, end};
    }
    public static String toggleCase(String input) {
        StringBuilder toggled = new StringBuilder();
        for (char ch : input.toCharArray()) {
            if (Character.isUpperCase(ch)) {
                toggled.append(Character.toLowerCase(ch));
            } else {
                toggled.append(Character.toUpperCase(ch));
            }
        }
        return toggled.toString();
    }
    public static void main(String[] args) {
        String input1 = "John";
        String input2 = "Johny";
        String input3 = "Janardhan";
         String[] parts1 = splitString(input1); // front, middle, end
        String[] parts2 = splitString(input2);
        String[] parts3 = splitString(input3);
         String output1 = parts1[0] + parts2[1] + parts3[2];
        String output2 = parts1[1] + parts2[2] + parts3[0];
        String output3 = parts1[2] + parts2[0] + parts3[1];
        output3 = toggleCase(output3);
        System.out.println("Output1: " + output1);
        System.out.println("Output2: " + output2);
        System.out.println("Output3: " + output3);
    }
}
```

Output:



```
input1 = "John"
input2 = "Johny"
input3 = "Janardhan"
```



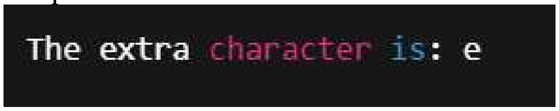
```
Output1: Jhhan
Output2: ohnyJan
Output3: NJOARD
```

Problem 4.

CODE:

```
public class ExtraCharFinder {
    public static char findAddedChar(String s, String t) {
        char result = 0;
        for (char ch : s.toCharArray()) {
            result ^= ch;
        }
        for (char ch : t.toCharArray()) {
            result ^= ch;
        }
        return result;
    }
    public static void main(String[] args) {
        String s = "abcd";
        String t = "abcde";
        char extraChar = findAddedChar(s, t);
        System.out.println("The extra character is: " + extraChar);
    }
}
```

Output:



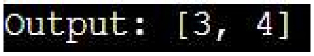
```
The extra character is: e
```

Problem 10.

CODE:

```
public class FindFirstAndLastPosition {
    public static int[] searchRange(int[] nums, int target) {
        int first = findBound(nums, target, true);
        int last = findBound(nums, target, false);
        return new int[]{first, last};
    }
    private static int findBound(int[] nums, int target, boolean isFirst) {
        int left = 0, right = nums.length - 1;
        int result = -1;
        while (left <= right) {
            int mid = left + (right - left) / 2;
            if (nums[mid] == target) {
                result = mid;
                if (isFirst) {
                    right = mid - 1;
                } else {
                    left = mid + 1;
                }
            } else if (nums[mid] < target) {
                left = mid + 1;
            } else {
                right = mid - 1;
            }
        }
        return result;
    }
    public static void main(String[] args) {
        int[] nums = {5, 7, 7, 8, 8, 10};
        int target = 8;
        int[] result = searchRange(nums, target);
        System.out.println("Output: [" + result[0] + ", " + result[1] + "]");
    }
}
```

Output:



```
Output: [3, 4]
```

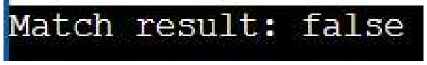
Problem 9.

CODE:

```
public class WildcardMatcher {
public static boolean isMatch(String s, String p) {
    int m = s.length();
    int n = p.length();
    // dp[i][j] means s[0..i-1] matches p[0..j-1]
    boolean[][] dp = new boolean[m + 1][n + 1];


    dp[0][0] = true;

    // Fill the first row for patterns like "*", "**", etc.
    for (int j = 1; j <= n; j++) {
        if (p.charAt(j - 1) == '*')
```

```
            dp[0][j] = dp[0][j - 1];
        }

        // Fill the DP table
        for (int i = 1; i <= m; i++) {
            for (int j = 1; j <= n; j++) {
                char sc = s.charAt(i - 1);
                char pc = p.charAt(j - 1);

                if (pc == '?' || pc == sc) {
                    dp[i][j] = dp[i - 1][j - 1];
                } else if (pc == '*') {
                    dp[i][j] = dp[i][j - 1] || dp[i - 1][j];
                }
            }
        }

        return dp[m][n];
    }

    public static void main(String[] args) {
        String s = "aa";
        String p = "a";

        boolean result = isMatch(s, p);
        System.out.println("Match result: " + result); // Output: false
    }

    }
```

Output:



```
Match result: false
```