

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n ^ 2 + m) time | O(n + m) space
5     func patternMatcher(_ pattern: String, _ string: String) -> [String] {
6         if pattern.count > string.count {
7             return []
8         }
9
10        let oldPattern = pattern.map { String($0) }
11        let newPattern = generateNewPattern(pattern)
12        let didSwitch = oldPattern[0] != newPattern[0]
13
14        var counts = ["x": 0, "y": 0]
15        let firstYPosition = populateCountsAndGetFirstYPosition(&counts, string)
16
17        if counts["y"] != 0 {
18            for lengthOfX in 1 ..< string.count {
19                if let countsOfX = counts["x"], let countsOfY = counts["y"] {
20                    let lengthOfY: Double = Double(string.count - (lengthOfX * countsOfX))
21
22                    if lengthOfY <= 0 || lengthOfY.truncatingRemainder(dividingBy: countsOfY) != 0 {
23                        continue
24                    }
25
26                    let indexOfY = lengthOfX * firstYPosition
27
28                    let startX = string.index(string.startIndex, offsetBy: lengthOfX)
29                    let endX = string.index(string.startIndex, offsetBy: lengthOfX * countsOfX)
30                    let x = String(string[startX ..< endX])
31
32                    let startY = string.index(string.startIndex, offsetBy: indexOfY)
33                    let endY = string.index(string.startIndex, offsetBy: indexOfY + lengthOfY)
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func patternMatcher(_ pattern: String, _ string: String) -> [String] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     func patternMatcher(_ pattern: String, _ string: String) -> [String] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

```
1 class Program {
2     func patternMatcher(_ pattern: String, _ string: String) -> [String] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

