

Our Solution(s)Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(w * n * log(n)) time | O(wn) space - where w is the number of words
5     // n is the length of the longest word
6     func groupAnagrams(_ words: [String]) -> [[String]] {
7         var anagrams = [String: [String]]()
8
9         for word in words {
10             let sortedWord = String(word.sorted())
11             if var arr = anagrams[sortedWord] {
12                 arr.append(word)
13                 anagrams[sortedWord] = arr
14                 continue
15             }
16             anagrams[sortedWord] = [word]
17         }
18
19         var result = [[String]]()
20         for group in anagrams {
21             result.append(group.value)
22         }
23         return result
24     }
25 }
26
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

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

```

14     return head[0]
15
16     program = Program()
17     headless("Test case 2")
18     # Create a new node
19     node = Node(10)
20     # Append the node to the list
21     program.append(node)
22     # Print the list
23     print(program.print_list())
24
25
26     headless("Test case 3")
27     # Create a new node
28     node = Node(10)
29     # Append the node to the list
30     program.append(node)
31     # Print the list
32     print(program.print_list())
33
34
35     headless("Test case 4")
36     # Create a new node
37     node = Node(10)
38     # Append the node to the list
39     program.append(node)
40     # Print the list
41     print(program.print_list())

```

Run or submit code when you're ready.