

Reducing dishes code.java

```
1  package Com.wipro;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Practice73 {
7      public static int calculateMaxCombinationSum(int[] array) {
8          int maxSum = Integer.MIN_VALUE;
9          List<List<Integer>> combinations = generateSubArrayCombinations(array);
10         for (List<Integer> combination : combinations) {
11             int sum = calculateSum(combination);
12             maxSum = Math.max(maxSum, sum);
13         }
14         return maxSum;
15     }
16
17     public static int calculateSum(List<Integer> numbers) {
18         int sum = 0;
19         int j = 1;
20         for (int number : numbers) {
21             sum += number * j;
22             j++;
23         }
24         return sum;
25     }
26
27     public static List<List<Integer>> generateSubArrayCombinations(int[] array) {
28         List<List<Integer>> combinations = new ArrayList<>();
29         backtrack(array, 0, new ArrayList<>(), combinations);
30         return combinations;
31     }
32
33     private static void backtrack(int[] array, int index, List<Integer> current, List<List<Integer>> combinations) {
34         combinations.add(new ArrayList<>(current));
35         for (int i = index; i < array.length; i++) {
36             current.add(array[i]);
37             backtrack(array, i + 1, current, combinations);
38             current.remove(current.size() - 1);
39         }
40     }
41
42     public static void main(String[] args) {
43         int[] array = { -1, -9, 0, 5, -7 };
44         System.out.println("Max Sum: " + calculateMaxCombinationSum(array));
45     }
46 }
47
```