

Chapter 5 Programming

```
34 public static void randomizeInPlace (int[] A) {  
35     int n = A.length;  
36     Random random = new Random();  
37     for (int i = 0; i < n; i++) {  
38         int rand = i + random.nextInt(n-i);  
39         int temp = A[i];  
40         A[i] = A[rand];  
41         A[rand] = temp;  
42     }  
43 }  
44  
45 public static void permuteWithAll (int[] A) {  
46     int n = A.length;  
47     Random random = new Random();  
48  
49     for (int i = 0; i < n; i++) {  
50         int rand = random.nextInt(n);  
51         int temp = A[i];  
52         A[i] = A[rand];  
53         A[rand] = temp;  
54     }  
55 }  
56 private static String arrayToString(int[] items) {  
57     StringBuilder sb = new StringBuilder();  
58     for (int item : items) {  
59         sb.append(item);  
60     }  
61     return sb.toString();  
62 }  
63 }
```

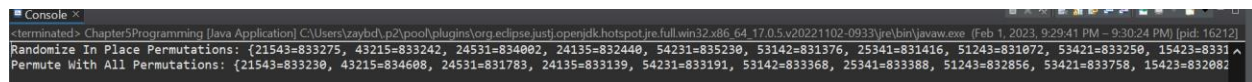
This code is fairly straightforward. The `permuteWithAll` and `randomizeInPlace` methods are practically identical, with the only difference being that `randomizeInPlace` will not pick previous values to swap with. I used the `arrayToString` method to make a string hashmap more easily.

```

4
5 public class Chapter5Programming {
6
7     static int numberOfTests = 100000000;
8
9     public static void main(String[] args) {
10         int[] items = {1, 2, 3, 4, 5};
11
12
13         HashMap<String, Integer> RIPpermutations = new HashMap<>();
14         HashMap<String, Integer> PWApermutations = new HashMap<>();
15
16         for (int i = 0; i < numberOfTests; i++) {
17             int[] tempItems = items;
18             randomizeInPlace(tempItems);
19             String permutation = arrayToString(tempItems);
20             RIPpermutations.put(permutation, RIPpermutations.getOrDefault(permutation, 0) + 1);
21
22             tempItems = items.clone();
23             permuteWithAll(tempItems);
24             permutation = arrayToString(tempItems);
25             PWApermutations.put(permutation, PWApermutations.getOrDefault(permutation, 0) + 1);
26         }
27
28         System.out.println("Randomize In Place Permutations: " + RIPpermutations);
29         System.out.println("Permute With All Permutations: " + PWApermutations);
30
31         HashMapToCSV.writeToCSV(RIPpermutations, "randomizePermutaions.csv");
32         HashMapToCSV.writeToCSV(PWApermutations, "permutePermutaions.csv");
33     }

```

The for loop allows us to iterate over the numberOfTests, perform both permutation methods, and add them into their respective hash map. I asked chatGPT the best way to add each string into the hashmap and this was the answer it gave. I find it to be quite nice, personally.



```

<terminated> Chapter5Programming [Java Application] C:\Users\zaybd\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.5.v20221102-0933\jre\bin\javaw.exe (Feb 1, 2023, 9:29:41 PM - 9:30:24 PM) [pid: 16212]
Randomize In Place Permutations: {21543=833275, 43215=833242, 24531=834002, 24135=832440, 54231=835230, 53142=831376, 25341=831416, 51243=831072, 53421=833250, 15423=83331
Permute With All Permutations: {21543=833230, 43215=834608, 24531=831783, 24135=833139, 54231=833191, 53142=833368, 25341=833388, 51243=832856, 53421=833758, 15423=832082

```

The randomizeInPlace is slightly more equal, in terms of how many of each permutation it produced. permuteWithAll, on the other hand, was much less consistent.

I didn't want to screenshot all of the results, but even with just that, its easily visible.

Below I have some graphs to illustrate each, and randomize is ever so slightly more consistent.

