Our Solution(s) Run Code

Solution 1 Solution 2

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using System.Collections.Generic;

public class Program {
  // O(n*n!) time | O(n*n!) space
  public static List<List<int> > GetPermutations(List<int> array
    List<List<int> > permutations = new List<List<int> >();
    GetPermutations(0, array, permutations);
    return permutations;
  }

  public static void GetPermutations(int i, List<int> array, Li
    if (i == array.Count - 1) {
      permutations.Add(new List<int>(array));
    } else {
      for (int j = i; j < array.Count; j++) {
        swap(array, i, j);
        GetPermutations(i + 1, array, permutations);
        swap(array, i, j);
      }
    }
  }

  public static void swap(List<int> array, int i, int j) {
    int tmp = array[i];
    array[i] =  array[j];
    array[j] =  tmp;
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
using System.Collections.Generic;

public class Program {
  public static List<List<int> > GetPermutations(List<int> array
    // Write your code here.
    return null;
  }
}

```

Our Tests

Custom Output Submit Code

```
using System.Linq;
using System.Collections.Generic;

public class Program{Test} {
  [Test]
  public void Test0(){
    int[][] input = new int[][]{};
    int[][]int[][] = parse + Program.GetPermutations(input);
    While.Assert(parse.Count == 0);
  }

  [Test]
  public void Test1(){
    int[][] input = new int[][]{
      {
      },
    };
    int[][]int[][] = parse + Program.GetPermutations(input);
    While.Assert(parse.Count == 1);
    While.Assert(int[][]equals(parse, new int[][]{
      {
      },
    }));
```

Run or submit code when you're ready.