

Our Solution(s)Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using System.Collections.Generic;
4
5 public class Program {
6     // O(n*n!) time | O(n*n!) space
7     public static List<List<int> > GetPermutations(List<int> array)
8     {
9         List<List<int> > permutations = new List<List<int> >();
10        GetPermutations(0, array, permutations);
11        return permutations;
12    }
13
14    public static void GetPermutations(int i, List<int> array, List<List<int> > permutations)
15    {
16        if (i == array.Count - 1) {
17            permutations.Add(new List<int>(array));
18        } else {
19            for (int j = i; j < array.Count; j++) {
20                swap(array, i, j);
21                GetPermutations(i + 1, array, permutations);
22                swap(array, i, j);
23            }
24        }
25    }
26
27    public static void swap(List<int> array, int i, int j) {
28        int tmp = array[i];
29        array[i] = array[j];
30        array[j] = tmp;
31    }
32 }
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 using System.Collections.Generic;
2
3 public class Program {
4     public static List<List<int> > GetPermutations(List<int> array)
5     {
6         // Write your code here.
7         return null;
8     }
9 }
```

```

1  #!/usr/bin/env python
2  # coding: utf-8
3
4  # Import the necessary libraries
5  import sys
6  import argparse
7
8  # Define the main function
9  def main():
10     # Create the parser
11     parser = argparse.ArgumentParser()
12     # Add the arguments
13     parser.add_argument('--input', type=str, required=True, help='Input file')
14     parser.add_argument('--output', type=str, required=True, help='Output file')
15     parser.add_argument('--format', type=str, required=True, help='Output format')
16     # Parse the arguments
17     args = parser.parse_args()
18
19     # Read the input file
20     with open(args.input, 'r') as f:
21         data = f.read()
22
23     # Process the data
24     # ...
25
26     # Write the output
27     with open(args.output, 'w') as f:
28         f.write(data)
29
30     # Print the output format
31     print(args.format)
32
33 # Call the main function
34 if __name__ == '__main__':
35     main()

```

Run or submit code when you're ready.