

**1-4\_Factors/1-4\_Factors\_Code.cs**

```
using System;
using System.Threading.Tasks; // Parallel.XXYYZZ()
// Add any using's here

class Factors {
    public static void Main(string[] none) {
        string first_string = Console.ReadLine()?.Trim() ?? string.Empty;
        string second_string = Console.ReadLine()?.Trim() ?? string.Empty;
        int start;
        int end;
        List<int>[] all_factors;// = new List<int>[]

        start = int.Parse(first_string);
        end = int.Parse(second_string);

        all_factors = new List<int>[end - start + 1];
        //Console.WriteLine(all_factors.Length);

        Parallel.For(start, end+1, i =>
        {
            //Console.WriteLine(i);
            all_factors[i-start] = GetFactors(i);
        });

        int i;
        string factors_string;
        for (i = 0; i < all_factors.Length; i+=1){
            Console.Write($"{i + start,5}: ");

            var factors = all_factors[i];
            factors_string = String.Join(' ', factors);
            /*
            foreach (var c in number) {
                Console.Write($"{c} ");
            }
            */
            Console.WriteLine(factors_string);
        }
    }

    public static List<int> GetFactors(int num) {
        List<int> factors = new List<int>();
        int i;
        for (i = 1; i < num+1; i += 1){
            if (num % i == 0){
                factors.Add(i);
            }
        }
    }
}
```

```
    }  
    return factors;  
}  
  
}
```