# yield IEnumerator

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;

public class Primes {
private long min;
private long max;

public Primes(long minimum, long maximum) {
min = minimum;
max = maximum;
}

public IEnumerator GetEnumerator() {
for (long possiblePrime = min; possiblePrime <= max; possiblePrime++) { bool isPrime = true; for (long possibleFactor = 2; possibleFactor <= (long)Math.Floor(Math.Sqrt(possiblePrime)); possibleFactor++) { long remainderAfterDivision = possiblePrime % possibleFactor; if (remainderAfterDivision == 0) { isPrime = false; break; } } if (isPrime) { yield return possiblePrime; } } } } class Program { static void Main(string[] args) { Primes primesFrom2To1000 = new Primes(2, 1000); foreach (long i in primesFrom2To1000) Console.Write("{0} ", i); } } [/csharp]
```