Filtering Primes

Scenario

Primes are numbers greater than 1, which are only divisible by 1 and the number itself. For instance, 2, 5, 7, 137 etc. are prime numbers while 4, 9, 12 are not since they can be divided by numbers other than 1 and 4, 9, 12 respectively.

Now **write a function** that takes a list of numbers (`integer array`), a list to keep filtered prime numbers (`integer array`) and the length of both the arrays (they need to be the same) as arguments. The function filters out the primer numbers from the first argument list and then puts them in the filtered list in the 2nd argument. Then the function returns the count of primes found in the first list. The declaration of the function looks like below:

```
/// @brief A function to filter out the prime number from a given list to a new list
/// @param unfilteredList A const int array containing unfiltered numbers
/// @param filteredList An int array that will hold the filtered prime numbers
/// @param length The length of the `unfilteredList` and `filteredList`
/// @return The size of the filtered prime list (count of prime numbers)
int FilterPrimes(const int* unfilteredList, int* filteredList, int length);
```

In the comment before the function, `@brief` is a brief description of what the function does. `@param <parameter_name>` is used to describe a specific parameter of the function. `@return` is used to describe the return values of the function.

Input

You may use any integer array with both prime and non-prime numbers to feed to the function and check for output. As a test case, you may use:

```
                {1, 2, 5, 4, 3, 6, 7, 11, 121, 119, 13, 10, 15},
a list of 13 numbers. You will find 6 prime numbers here,
                              {2, 5, 3, 7, 11, 13}.
```

Output

You may print out the count of prime numbers, as well as the elements of `filteredList` array to check whether all the primes from `unfilteredList` are present there or not.

Hint

You may define another function, `int IsPrime(int num)`*, to make your work easier and code cleaner.*