# Assignment 8

4 November 2012

**Due Monday 12 November at 1am**

In this exercise, we will print a figure on the screen representing the **prime** numbers. A prime number is a number with exactly two *factors:* one, and itself. A factor is a number that can divide the integer with no remainder.

For example, the only factors of 17 are 1 and 17. Nothing else can divide 17 without remainder, so 17 is prime. In contrast, the number 12 has lots of factors. It can be divided by 1, 2, 3, 4, 6, and 12. Therefore, 12 is not prime.

Do all this work in `a08/primes.cpp`, please.

## Testing for primes

Begin by thinking about how a computer program could determine whether a number is prime. You know how to loop through all the integers in a range, and you know how to test divisibility (modulo operator), so compose those together and you can get a program that does this:

```
Enter a number: 17
17 IS prime.

Enter a number: 12
12 is NOT prime.
```

Of course, your technique should work for other primes and non-primes too. For reference, here's a list of the first 20 primes: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71

## Printing the primes

Once you have the above working, rename `main` as `testPrimes` and then create a new `main` function for this section of the assignment.

Now, you will apply a prime test in a loop, so that your program can print out all the prime numbers up to some limit.

```
Enter a limit: 100
2 3 5 7 11 13 17 19 23 29
31 37 41 43 47 53 59 61 67 71
```

```
73 79 83 89 97 101 103 107 109 113
127 131 137 139 149 151 157 163 167 173
179 181 191 193 197 199 211 223 227 229
233 239 241 251 257 263 269 271 277 281
283 293
```

If you have some extra time, use `setw` to make the format of this list a little nicer:

```
Enter a limit: 100
    2     3     5     7    11    13    17    19    23    29
   31    37    41    43    47    53    59    61    67    71
   73    79    83    89    97   101   103   107   109   113
  127   131   137   139   149   151   157   163   167   173
  179   181   191   193   197   199   211   223   227   229
  233   239   241   251   257   263   269   271   277   281
  283   293
```

## Using primes to design a figure

Once you have the above working, rename `main` as `listPrimes` and then create a new `main` function for this section.

Now, we can use primes to design a text figure with interesting patterns. The user can specify the number of rows and columns, and then the program will print out stars to represent primes, or dashes to represent non-primes.

The cells in the table correspond to the integers starting from 1 in the first row, and then continuing on the next row. Here are some examples:

```
How many rows? 8
How many columns? 24
-**-*-*---*-*---*-*---*-
----*-*-----*---*-*---*-
----*-----*-*-----*---*-
*-----*---*-----*-------
*---*-*---*-*---*-------
------*---*-----*-*-----
----*-*-----*-----*---*-
----*-----*-*---------*-
```

```
How many rows? 20
How many columns? 37
-**-*-*---*-*---*-*---*-----*-*-----*
---*-*---*-----*-----*-*-----*---*-*-
----*---*-----*-------*---*-*---*-*--
```