**AP Computer Science - Prime Day**

**Program # 1 - Emirp**

An emirp is a non-palindromic prime number whole reversal is also a prime. For example, 17 and 71 are both prime and are reversals of one another, so they are emirps. Write a program to display the first 100 emirps.

**Program # 2 - MersennePrime**

A prime number is called a Mersenne prime if it can be written in the form 2n - 1 for some positive integer n. Write a program that finds all Mersenne primes with n <=31 and displays the output in a table format:

n 2^n-1

2 3

3 7

5 31

**Program # 3 - TwinPrimes**

Twin primes are prime numbers that differ by only 2. For example, 3 and 5 are TwinPrimes, as are 5 and 7. Write a program that will find all twin primes less than 1000. Display the twin primes in the following format, one per line:

(3, 5)

(5, 7)

….

**Reminder**

A number is prime if it is divisible by only 1 and itself. Alternately, we can start at 2 and stop one shy of the number to be checked and if we find any number that is a factor (mod function), we can determine it is not prime. If we find no factors for given number, *num*, from 2 to *num*-1, we can declare it as prime. Below is the syntax for a usable prime calculation method. It can, of course, be used in all three of the above problems.

**public** **static** **boolean** isPrime(**int** num)

 {

 **for** (**int** x =2; x<num; x++)

 {

 **if** (num %x==0) **return** **false**;

 }

 **return** **true**;

 }