HW3 (Due 2018/03/26)

1. An integer is said to be a **perfect number** if the sum of its divisors, including 1 (but not the number itself), is equal to the number. For example, 6 is a perfect number, because 6 = 1+2+3. Write a C++ program to print all the prefect numbers between 2 to *n* (<= 10000), and list the divisors of each perfect number.

   Example : (The italics for program output and boldfaces for user input)

Please input an integer n: **1000** *Perfect number(s) between 2 and 1000:* *6 = 1 + 2 + 3* *28 = 1 + 2 + 4 + 7 + 14* *496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248*

   Bonus:

   Write a function ***int IsPrefect* (*int number*)** to determine whether parameter ***number*** is a perfect number. The function returns 1 if ***number*** is a perfect number, and 0 otherwise. Use this function to complete your program.

(1) For *n* in $\mathbf{Z}^+$, *n* >= 2, the Euler's phi function $\phi(n)$ is the number of positive integers *m*, where 1<= *m* < *n* and gcd(*m*,*n*) = 1. For example, $\phi(2)$=1, $\phi(3)$= 2, $\phi(4)$ = 2, $\phi(5)$ = 4 and $\phi(6)$ = 2. Write a C++ program to find $\phi(n)$ for a given *n*, where 2 <= *n* <= 32767, and list the integers *m*, where 1<= *m* < *n* and gcd(*m*,*n*) = 1

   Example : (The italics for program output and boldfaces for user input)

Please input an integer n: **6** *Euler's phi function of 6 is 2,* *where gcd(6,1)=1 and gcd(6,5)=1.*

   Bonus:

   Write a function ***int IsPrimeToEach(int m, n)*** which returns 1 if gcd(*m*,*n*) = 1 and 0 if gcd(*m*,*n*) $\neq$ 1. Use this function to complete your program.