### **SIENA COLLEGE**

**29th Annual** High School Programming Contest

##### **April 1, 2016**

###### Problem #3:  Too abundant, too deficient, or just perfect!

Background Information:  A positive integer N is said to be a *perfect number* if the sum of the proper factors of N is equal to N. A proper factor of N is a positive integer less than N that divides N. For example, the proper factors of 12 are 1, 2, 3, 4, and 6. If the sum of the proper factors of N is greater than N, N is an *abundant number.* For example, 12 is an abundant number because 1+2+3+4+6 = 15 which is greater than 12. If the sum of the proper factors is less than N, N is a *deficient number*. Ten is an example of a deficient number because the proper factors of 10 are 1, 2, and 5 and 1 + 2 + 5 = 8 < 10. The smallest perfect number is 6. Six is a perfect number because the proper factors are 1, 2, and 3 and 1 + 2 + 3 = 6. Note that 1 is a deficient number.

Note: For this problem, your program will need to be efficient enough to compute the desired results in less than one minute of CPU time. If your program takes longer than a minute to run, you should think about ways you can make it faster.

###### Programming Problem:

Input:  A pair of non-negative integers A and B with A ≤ B ≤ 1,000,000.

 The A and B will be input on two lines.

Output: The values of A and B followed by the number of perfect, abundant, and deficient

numbers A and B inclusive. These five integers must be output on five consecutive lines.

###### Example : Input:  1

250

######  Output:  1

###### 250

###### 2

###### 57

###### 191