

Testing Report for

Sum Of Factors

Prepared by Samantha Superprogrammer

The Sum Of Factors program claims to do exactly what its name implies – to sum all of the factors of a given input value. The program contract states that valid inputs are in the range 200 to 2,000,000.

The test suite initially contained the values 231 (because it was the exemplar in the contract) as well as 200 and 2,000,000 (because they were the limits of the input range.) Analysis of the problem suggests that the following types of data might be of interest:

- 1, because the issue of “factoring” 1 is semantically tricky
- A prime number
- A product of exactly two primes (common in cryptography)
- A power of primes
- A product of two prime squares
- A product of two primes each greater than 200 (in case the lower input limit was a different kind of limit)

Of these, the first is ruled out as it represents an improper input for this particular program. Thus, the initial test suite – with anticipated answers was:

Input value	Sum of Factors
231	384
200	465
2000000	4980405
733	734
481 (=37*13)	532
4913 (=17 ³)	5220
1030225 (= 5 ² *203 ²)	1539057
43931 (=223*197)	44352

The program was tested on each of these values the correct answer was returned in each case *except for 1,030,225*. More testing was undertaken in this case. First, another exemplar was tested, and it also produced an incorrect answer. Then a single perfect square was tested, and it too gave the wrong answer. Several more perfect squares were tested and all produced incorrect results. A summary is given in the table on the next page:

Input value	Correct Answer	Answer Given	Difference
1030225 ($= 5^2 \cdot 203^2$)	1539057	1540072	1015
20449 ($= 11^2 \cdot 13^2$)	24349	24482	143
289 ($= 17^2$)	307	324	17
10201 ($= 101^2$)	10303	10404	101
537289 ($= 733^2$)	538023	538756	733

It is interesting to note that in each case, the difference in the two answers is exactly the square root of the input number. This strongly suggests that the square root of a perfect square is being ‘double counted’ in some way. **This is either a bug in the program – or, if this is the desired behavior, an error in the contract.**

Finally, we note that other values, outside the range were tested. The contract specifies that no such values will ever be given; therefore any behavior is acceptable in these circumstances. We tested integers outside the range – including negative numbers, 0, 1, 199, and 2000001; decimal numbers and even alphanumerical strings. In each case, the program produced an error message in a dialog box and continued gracefully.