###

**36th Annual High School Programming Contest**

| **Sponsored by** |  |
| --- | --- |

##### April 12, 2024

###### Green Problem #1: Some Efficient Summing:

Background Information:

Computer scientists frequently need to analyze the efficiency of their algorithms. Software that includes inefficient code can result in poor run-time performance. Imagine an internet search that requires years of computer time. This would be unacceptable.

Consider this algorithm: If the input for M is 3 and the input for N is 6, then Sum will

Input M be incremented 18 times and the output of Sum will be 18.

Input N By carefully tracing this algorithm, one can see that

Sum ← 0 when A = 3, the inside loop will be executed 3 times,

For A ← M to N Do when A = 4, the inside loop will be executed 4 times,

 For B ← 1 to A Do when A = 5, the inside loop will be executed 5 times, and

 Sum ← Sum + 1 when A = 6, the inside loop will be executed 6 times.

Output Sum So Sum will be incremented 3 + 4 + 5 + 6 = 18 times.

A faster algorithm to add A still faster algorithm is:

consecutive integers is:

Input M Input M

Input N Input N

Sum ← 0 Sum ← $\frac{N^{2} + N - M^{2} + M^{}}{2}$

For A ← M to N Do Output Sum

 Sum ← Sum + A

Output Sum

###### Programming Problem:

Input:  Integers M and N on separate lines with -10000 ≤ M < N ≤ 10000.

Output: The sum of the integers from M to N.

Example 1: Input: Example 2: Input: Example 3: Input:

 3 -5 1

 6 8 100

 Output: Output: Output:

 18 21 5050