Example 1 – Adding a series of numbers

Add the following set of numbers:
17, 20, 1, 22, 18

Step 1 in the problem solving sequence states:

"Understand the problem statement"

The purpose of the algorithm is to add together the above five numbers to calculate their combined total.

Step 2 in the problem solving sequence states:

"Develop an algorithm to solve the problem"

The **inputs** to the system are the numbers: 17, 20, 1, 22, and 18.

These inputs can be represented by the symbols:

n_1: 17
n_2: 20
n_3: 1
n_4: 22
n_5: 18

The **output** of the system is the calculated sum of the input numbers.
sum = 17 + 20 + 1 + 22 +18

Solution steps:

1. start with the sum equal to zero
2. take the first number
3. add the first number to the sum
4. get the second number
5. add the second number to the sum
6. get the third number
7. add the third number to the sum
8. get the fourth number
9. add the fourth number to the sum
10. get the last number
11. add the last number to get the final sum

The above 11 steps will produce the desired answer, in this case, the sum of the five numbers. However, steps two through eleven appear to be repetitive. Can the above steps be combined and reduced???

If the set of numbers are referred to by the above listed indexes n_1 through n_5 then the above solution can be re-written. If the letter i is used to replace the numbers 1 through 5 for the indexes, the algorithm can then be written in terms of n_i. Applying the above information provides the following:

Solution steps:

1. sum = 0
2. i = 0
3. i = i + 1
4. sum = sum + n_i
5. repeat steps 3 and 4 until i = 5

The above steps will produce the same sum for the five given numbers as the previous eleven steps. If the two algorithms produce the same result, what's the difference? Neither algorithm is necessarily *wrong* but the second algorithm tends to be more structured. Since this is a lesson in algorithm development for structured programming, the second algorithm would be the preferred one of the two. The exact reasons for selecting the second algorithm fall under the implementation phase of problem solving. Since this section is only concerned with developing possible solutions, we will leave this conversation for later. This example will be re-visited later during the lessons on flowcharting and control statements.

Recap:

Looking back to the previous description of an algorithm:

> The inputs to the algorithms are the actual numbers n_1 through n_5 found in the data set. The output of the algorithm is the calculated sum of the numbers. The steps necessary to solve this algorithm are stated precisely (steps 1 through 5). Each step in the algorithm produces a unique value. The algorithm works for the given set of numbers (finite, ends after five numbers). The algorithm is also general in nature, meaning that the steps will work for any set of numbers.