1. Does your algorithm have defined inputs and outputs?

Task 1 -

Yes. The question states that the input is [5,3,8,6,1,9,2,7], however the user is able to chose a different number of elements and values for their input. The output on the other had is not defined. We know the same integers will be output but we do not know what order the values will be in.

Task 2 –

There are no defined inputs and outputs for this program. The user is able to input any integer they would like and the program should work correctly for every value. The program performs the same calculation regardless of the value, therefore you could work out the expected outcome if you wish.

1. Can you guarantee that it terminates?

Task 1 – The program is not guaranteed to terminate. If for example the user enters a string of characters when the computer was expecting an integer then you would receive a TypeError. This is one example of where the program would not terminate. Likewise if you do not enter a value at all then you would also receive an error message.

Task 2 – similarly to the first task, if the user inputs a value with the incorrect data type then they will receive a type error. There is also a possibility that the program runs but outputs the incorrect value if some of the mathematical calculations are incorrect. This would result in a logic error which will not prevent the program from terminating but will return the incorrect results from the calculation. No error message is displayed for a logic error as the syntax of the program is correct.

1. Is it specified in a clear and concise manner?

Task 1 – I have tried to use good code practice in three ways. Firstly I have commented my code so that if I or another person opens up this file then they can see what each function or calculation does. In addition to this I have used functions to separate blocks of code, this helps to make the code more readable and allows programmers to reuse these functions in the future. Finally, I have used indentation to make it easier to read the flow of control and identify the statements that will and will not get executed during compilation.

Task 2 – just like task 1, I have been consistent with the code practices to help make my code efficient and easy to read. I believe I have included enough comments to identify the functions and the calculations that are performed.

1. Does your algorithm produce the correct result for all instances?

Task 1 – I have tested this program with a range of list sizes and values and it has worked correctly for all integer instances. However, when I enter the string data type for these numbers; i.e. ‘one’ instead of 1, I receive runtime errors as the program was expecting integers to be input. To correct this, I could write some if statements that compare the data types of the input data and the expected data and I could return some personalised error messages if the data has the incorrect type.

Task 2 – Likewise in this task, all of the integer input returns the correct values; this tells me that the calculations in the code are likely to be correct. However when any data type other than integers are input, error messages are returned. Jus like the first task, I can combat this by implementing if statements to check the data types of all input.