For Part 1:

To add up the two hexadecimal numerals, we would first have to convert every digit of the two hexadecimal numerals into decimal values. Then, by comparing the lengths of 'n1' and 'n2', we get three scenarios when adding up the two numerals:

1. When the number of digits that have been added up is less than the minimum length of 'n1' and 'n2': Add both the corresponding digit of 'n1' and 'n2'.
2. When the length of 'n1' is greater: Add the corresponding digit of 'n1'.
3. When the length of 'n2' is greater: Add the corresponding digit of 'n2'.

After the addition for each digit, set the carry to the sum of the digit divided by 16, and the final sum of the digit to the remainder of the sum after it has been divided by 16.

When the addition is complete, change the digits back to hexadecimal values and check if the carry is 0. If not, shift every digit of 'sum' to the right by one digit and set sum[0] to 1.

For Part 2:

To multiply the two numbers, we would once again require every digit of 'n1' and 'n2' to be turned into decimal values. Then, take a digit from 'n1' and multiply it with every digit in 'n2' each time until every digit in 'n1' has completed the process. The value of the production for each digit is added up in their corresponding position in 'Product'.

After that, for each digit, calculate the carry(Product[l1+l2-(i+2)]/16) and the correct value for each digit(Product[l1+l2-(i+2)]%16). When this process is completed, check if the carry is 0. If not, shift every digit of 'Product' to the right by one digit and set Product[0] to the value of carry. Lastly, convert the value of every digit of 'Product' to hexadecimal and store it in 'product'.

Printing out the results:

'Leading' (The length of product or sum +2) is used to calculate how many blanks should be printed out before the hexadecimal representation of 'n1' and 'n2'. Its value should be equivalent to the number of '-' that should be printed out. The number of blanks that should be printed before 'n1' is leading - length of 'n1', while that of 'n2' equals to leading - length of 'n2' - 2 (the subtraction of 2 eliminates the

two blanks that have already been occupied by '+' or '*' and a blank). For the sum or product, 2 blanks will be printed before it.