Programming Assignment 4: Hexadecimal Arithmetic

Part 1: Write a C program to add two hexadecimal numbers. The input data contains an even number of hexadecimal digit strings with maximum 64 bits and no leading zeros. The last two numerals are "0 0". Repeat the addition operation until both input hexadecimal numbers are 0's. In each iteration, the program will read two hexadecimal digit strings, n1 and n2, and add these two hexadecimal numbers with the result sum. In the output, print n1, n2, and sum aligning to the right with a "+" sign before n2 and a separated line below n2. Print "n1 + n2 = sum" in decimal. If the result exceeds 64 bits, print an overflow message. Verify correctness of the sum using decimal addition. Sample input (hexadecimal_addition.txt):

```
1 FFFF
C9AE0232 5F1300033
3DBCE123E EAD2222
2873383898 D93E8C2FA3C
5555555555555555 AAAAAAAAAAAAAAAA
FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF
0 0
```

Sample execution code (hexadecimal_addition.c). Program execution example:

```
命令提示字元
D:\>hexadecimal_addition < hexadecimal_addition.txt
      1
+  FFFF
-------
  10000
1 + 65535 = 65536

   C9AE0232
+ 5F1300033
-----------
  6BADE0265
3383624242 + 25521291315 = 28904915557

  3DBCE123E
+   EAD2222
-----------
  3EA7B3460
16572617278 + 246227490 = 16818844768

   2873383898
+ D93E8C2FA3C
-------------
  DBC5BFB32D4
173731756184 + 14928916445756 = 15102648201940

  5555555555555555
+ AAAAAAAAAAAAAAAA
-----------------
  FFFFFFFFFFFFFFFF
6148914691236517205 + 12297829382473034410 = 18446744073709551615

   FFFFFFFFFFFFFFFF
+  FFFFFFFFFFFFFFFF
------------------
  1FFFFFFFFFFFFFFFE
18446744073709551615 + 18446744073709551615 = 18446744073709551614    ****Overflow!!!
```

Part 2: Write a C program to multiply two hexadecimal numbers. The input data contains an even number of hexadecimal digit strings with maximum 32 bits and no leading zeros. The last two numerals are "0 0". Repeat the multiplication operation until both input binary

numbers are 0's. In each iteration, the program will read two hexadecimal digit strings, n1 and n2, and multiply these two hexadecimal numbers with the result product of maximum 64 bits. Note that the length of product could be the total length of n1 and n2. Hence, no overflow will occur. In the output, print n1, n2, and product aligning to the right with a "*" sign at the left-most character in the output line of n2 and a separated line below n2. Print "n1 * n2 = product" in decimal. Verify correctness of the product using decimal multiplication. Sample input (hexadecimal_multiplication.txt):

```
1 FFFF
34A C51
C9A32 5F0033
3DB3E21 EAD22
55555555 AAAAAAAA
FFFFFFFF FFFFFFFF
0 0
```

Sample execution code (hexadecimal_multiplication.c). Program execution example:

```
命令提示字元
D:\>hexadecimal_multiplication < hexadecimal_multiplication.txt
      1
*  FFFF
------
   FFFF
1 * 65535 = 65535

      34A
*     C51
--------
   28826A
842 * 3153 = 2654826

         C9A32
*       5F0033
-------------
   4AD3B10B7F6
825906 * 6225971 = 5142066804726

        3DB3E21
*         EAD22
--------------
   389911E88D62
64699937 * 961826 = 62230081604962

           55555555
*          AAAAAAAA
------------------
   38E38E3871C71C72
1431655765 * 2863311530 = 4099276458915470450

           FFFFFFFF
*          FFFFFFFF
------------------
   FFFFFFFE00000001
4294967295 * 4294967295 = 18446744065119617025
```

In this assignment, you must submit two source code files **assignment4_DXXXXXXX_1.c** (40%, for Part 1 hexadecimal addition) and **assignment4_DXXXXXXX_2.c** (40%, for Part 2 hexadecimal multiplication) and

one report file to explain how you solve the two problems and write the programs **assignment4_DXXXXXXX.pdf** (20%), where **DXXXXXXX** is your student ID. Programming assignment 4 is due by **23:59 pm, Sunday, November 26**. Submit your solution files and the report to **iLearn.**