1. [Large unsigned integer arithmetic]
You will represent an arbitrarily large unsigned integer *I* by a singly linked list of its "digits" d_1, d_2, ... , d_m, where d_1 is the least significant "digit" of *I* and d_m is the most significant "digit" of *I*. Assume a base of $10{,}000_{10}$. Implement each "digit" as a C *unsigned int* with "digit" values in [0, 9999].

a) Write C functions to perform the arithmetic operations addition and multiplication of large unsigned integers.
b) Write a C function to input a large unsigned integer from stdin as comma separated "digits" in decimal, ordered on decreasing significance and terminated by a $
c) Write a C function to output a large unsigned integer to stdout in the format used for input, showing each "digit" as 4 decimal digits.
d) Write a C main() which will repeatedly accept (from stdin) and evaluate, large integer infix expressions on large unsigned integer constants using the operators "+" (addition) and "*" (multiplication). An expression is terminated by "=". On encountering an "=", the value of the expression is to be printed to stdout. For simplicity, evaluate expressions from left to right.

For example, the expression

1111,0411,4111,1111,0011$ + 2222,2222$ * 0003$ =

results in output

3333,1234,2333,9999,6699$