You are to write two MIPS assembly language subroutines that perform `read_hex` and `print_hex` functions (similar to the `read_int` and `print_int` syscalls). The first routine will input a string representing a hexadecimal number (that is, a string consisting of the digits 0 - 9 and the characters A - F, a - f), and return the `unsigned` integer value represented by the string.

The second routine should take an unsigned number as an argument, produce a string consisting of the hexadecimal characters that represent the number, and output the string using the `print_str` syscall. You don’t have to suppress leading zeroes, although you can if you want (most output routines do so!).

To test your two routines, your main program should call the input routine twice, find the sum of the two numbers that are input, and output the result using your output routine. Your main program should continue to accept pairs of numbers and print their sum until two zero values are input. If an illegal character is input for a number, your program should print an error message, and then prompt for the number again.

Sample run:

Enter the first number: A3
Enter the second number: 2c
The sum of the numbers is: 000000CF

Enter the first number: 5t
***Illegal number!***
Enter the first number: fF17
Enter the second number: a4
The sum of the numbers is: 0000FFBB

Enter the first number: 0
Enter the second number: 0

Done!