CS061 – Programming Assignment 06

Objective

You may not know it, but in C++, you can use manipulators to alter the base of a number when displaying it on the console. The purpose of this assignment is to bring you to a deep understanding of how this kind of process works and to implement it yourself in LC3.

High Level Description

Build a decimal-to-hex converter as a subroutine, and build an appropriate test harness for it. The range of possible input numbers is [0, 65535].

Examples

Fun Fact:
In C++, the following will print the value of x in hexadecimal format:

```
int x = 12092;
cout<< hex << x;   // outputs “2f3c”
```

For this assignment:
User enters “#64222” , the program outputs “xFADE”
User enters “#10” , the program outputs “xA” or “x000A”
User enters “#2468” , the program outputs “x9A4” or “x09A4”

Your Tasks

This programming assignment can be broken into four steps:

1. The testbed asks the user to enter a decimal number between [0, 65535].
2. The testbed then calls the DEC_TO_HEX subroutine.
3. The DEC_TO_HEX subroutine translates the decimal number into a hexadecimal representation and stores it as a null-terminated string in an array.
4. The testbed prints out the array – i.e. the hexadecimal number.

- Zero-padded input is inappropriate at this point (you can do better!). The user should NOT be asked to input numbers in the form “#00025”.
- No error checking required – you may assume always valid input (yay!!☺)

Uh…help?

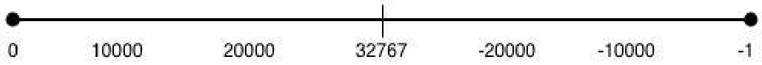
There are a number of things to consider for this assignment:

- The input can be anything up to #65535.
  Nut the LC3 can only handle 16-bit 2’s complement numbers, so any number that goes beyond 32767 will overflow to the negative range of numbers
  (for example: in LC3, 32767 + 1 == -32768 – *make sure you understand this!*).

- If the number entered is in the range [0,32767] then you can convert it the way you would do it on paper

- If the number entered is in the range of [32768, 65535], your algorithm will have to act in a slightly different way until, as you are translating, the number returns to the [0,32767] range.

- In order to perform the translation algorithm, it helps to reinvent the natural number line:



Now the "biggest" possible number is #-1 and the "smallest" possible number is #0. The standard paper-based algorithm for translating decimal numbers to 4-digit hexadecimal representation (which you already know) is:
Normal Decimal To Hex Translation Algorithm:
(1) How many 16^3 are there in this number? (that's the first hex digit)
(2) How many 16^2 are there in the remaining number? (that's the second hex digit)
(3) How many 16^1 are there in the remaining number? (that's the third hex digit)
(4) How many 16^0 are there in the remaining number? (that's the fourth hex digit)

This is nearly identical to Exercise 01 from Lab 06, where you took a number in a register and printed it out in base-10 (decimal).

The only difference between the "normal" algorithm and the algorithm you will write is that your algorithm will proceed in two steps:

(1) Modified part
(2) Normal part

For the modified part, instead of seeing how many 16^x you can *subtract* from the number until the remaining number becomes *negative*, you check how many times you can *add* 16^x before the remaining number becomes *positive*.

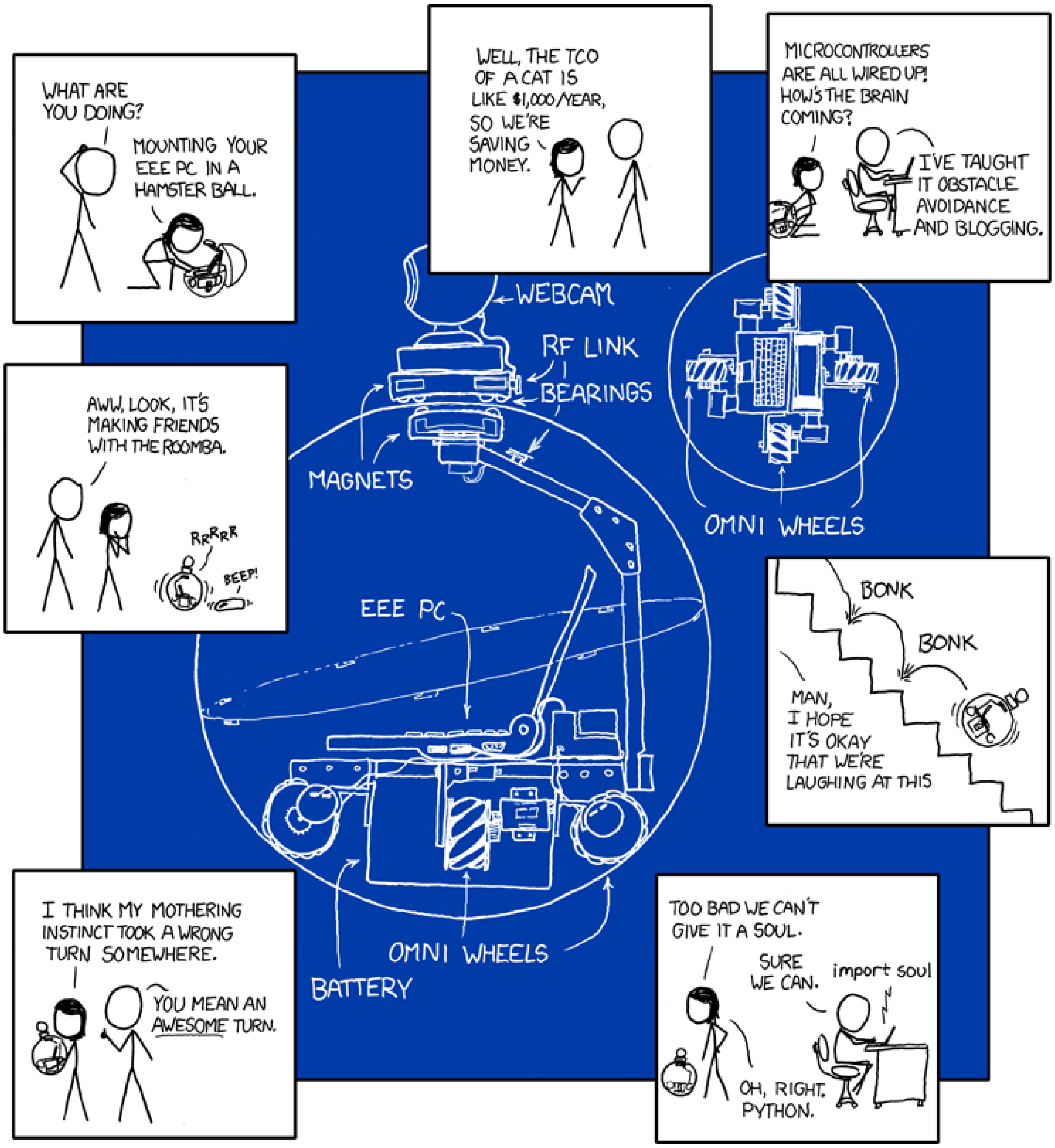
For the normal part, follow the normal paper-based algorithm for converting numbers to hex.

Rubric

- Code does not assemble: -10 points *(no reshow)*
- No header: -10 points *(no reshow)*
- Zero-padded input: (-2 points)
- Not following specifications: -??? points (depending on severity)

- Well commented code: +2 points
- Proper input of decimal number from user: +2 points
- Proper hex translation for numbers in the range [0, 32767]: +3 points
- Proper hex translation for numbers in the range [32768, 65535]: +3 points

Comics?!Sweet!!


WHAT ARE YOU DOING?
MOUNTING YOUR EEE PC IN A HAMSTER BALL.
WELL, THE TCO OF A CAT IS LIKE $1,000/YEAR, SO WE'RE SAVING MONEY.
MICROCONTROLLERS ARE ALL WIRED UP! HOW'S THE BRAIN COMING?
I'VE TAUGHT IT OBSTACLE AVOIDANCE AND BLOGGING.
WEBCAM
RF LINK
BEARINGS
MAGNETS
OMNI WHEELS
AWW, LOOK, IT'S MAKING FRIENDS WITH THE ROOMBA.
RRRRR
BEEP!
EEE PC
BONK
BONK
MAN, I HOPE IT'S OKAY THAT WE'RE LAUGHING AT THIS
BATTERY
OMNI WHEELS
I THINK MY MOTHERING INSTINCT TOOK A WRONG TURN SOMEWHERE.
YOU MEAN AN AWESOME TURN.
TOO BAD WE CAN'T GIVE IT A SOUL.
SURE WE CAN.
import soul
OH, RIGHT. PYTHON.

Source: http://xkcd.com/413/