# Homework – Print numbers 0 to *n-1*

This is a programming assignment that will be graded and scored as a homework since it is so simple.

Write a program for the LC3. This program should be named ***hwPrintN-1.hex.*** This hex file will be a machine language program for the LC3 processor.

**This program will be written as an ASCII hex file.**

**DON'T TRY TO ASSEMBLE IT WITH AN LC3 ASSEMBLER.**

The hex file must contain only hex characters; 0-9 and A-F with no x, X, 0x, or 0X.

Each line should be a four digit hex number. Do not include a 0x or x at the beginning of this number.

The first line should be the starting address in memory, use 3000 to be safe.

The last line should be a halt (F025)

The lines between the first and last line should do the following: Write a program for the LC3. This program should be named ***hwPrintN-1.hex.***

> Print a number of digits specified in R1.
>
> These digits should start with 0 and count up and then stop at R1 – 1.
>
> For example:
>
> If 5 is in R1 (the number 5 **NOT** the ASCII code for 5), your program would print 01234
>
> If 9 is in R1, (the number 9 **NOT** the ASCII code for 9) your program would print 012345678
>
> You can assume R1 will never be less than zero or greater than 10. You DO NOT have to do any type of error checking.
>
> DO NOT initialize R1 in your program. Assume the value will already be in R1 when your program starts.
>
> For testing, you should load your program into the simulator and then use the simulator to change R1 to some initial value by clicking on it and entering a value from 0 to 10.

**Write your program in assembly. Convert it manually to hex. Put the hex in *hwPrintN-1.hex*.**

Test your program using the simulator.

When you test your program, you will need to insert the number of digits into R1 before running your program. Load the program, put a number 0 to 10 in R1, then run your program.

**DO NOT INSERT THE NUMBER OF CHARACTERS INTO R1 WITH YOUR PROGRAM. THIS WILL CAUSE IT TO FAIL THE TESTS.**

**ONLY R1 should be manually entered. DO NOT MANUALLY ENTER THE ASCII CODE FOR 0 INT0 R0.**

Store the ASCII code for 0 below the halt command and load that value into R0.

You should not need and are not allowed to use ST in this program.

Submit your ***hwPrintN-1.hex*** file to Web-CAT. Make sure you did not name it ***hwPrintN-1.txt*** or that Notepad or WordPad didn't rename it ***hwPrintN-1.hex.txt***.

NOTE: You will lose 1 point for every submission over 5. You will lose 10 points for each day late. Submitting more than 2 days late will result in a zero.

Note that late days start at 9:00 am. So, submitting after 9:00 am on the due date will result in -10 points. Submitting after 9:00 am on the day after that will result in an additional -10 points. You will not be able to submit after 9:00 am on the day after that.

A possible algorithm.

```
Copy the ASCII code for zero to a register.  R0 would be easiest.
while (R1 indicates not done)
{
        print ascii
        ascii ++
        R1 --
 }
```

If R1 starts at 7, for example, and you decrement it every time through the loop, how do you know when you are done?

The equivalent to the above in LC3 assembly might look something like this.

```
Copy the ASCII code for zero to a register.  R0 would be easiest.
If R1 indicated done, branch to end.
        print ascii IN R0
        add 1 to ascii
        R1 --
Always branch back to if statement
```