Durham University

LMC Programming Assignment

Coursework Description and Assessment

The deadline for submission is the 9th November (2pm). You should submit assembly code (i.e. in the LMC simulator select **Save** from the LMC Assembly Editor window). You should submit the program as a plaintext file (.txt), with comments to indicate how its works. The filename of the program file should be BaseConverter.txt. **I should be able to open the file with the LMC Assembly Editor window and it should compile and run correctly without alteration!** Please don't zip files together - just attach plaintext .txt file to the DUO submission.

Objective
You should create a LMC program to covert numbers from one base to another. Your program should take three inputs: n, a, b and convert the number n from base a to base b. E.g. if the inputs are 101, 2, 3. This means convert the base 2 number 101 into base 3. The output should therefore be 12.

E.g.	Input 123, 10, 5	Output: 443
	Input 321, 4, 10	Output: 57
	Input 461, 8, 7	Output: 614

You program should correctly run a second time if the program counter is reset to zero at any time. As an advanced objective, your program should detect overflow whenever it occurs and output 999 to indicate this error. For example, 888 (base 9) is 2060 in base 7. Hence the program should behave as follows:

Input 888, 9, 7	Output: 999

Your program should handle all inputs elegantly, returning 999 whenever it cannot correctly compute the base conversion. If you have made a design choice as to how to handle a specific situation, please put a comment at the top of your code to explain the choice.

Test 1 (30%)
I will test the program on a variety of conversions into base 10.
E.g. Input 123, 5, 10
Input 011, 2, 10

Test 2 (10%)
I will test the program on a variety of conversions out of base 10.
E.g. Input 123, 10, 5
Input 006, 10, 2

Test 3 (10%)
I will test the program on a variety of conversions between bases.
E.g. Input 123, 5, 7
Input 011, 2, 4

Test 4 (25%)
I will test the program on a variety of tricky inputs.

A pass mark (40%) is possible for a program that correctly converts any base into base 10 (Test 1), and deals correctly with tricky cases when converting into base 10 (part of Test 4). You should ensure you program at least manages this.
An excellent mark (75) will be awarded for programs that give the correct answer in all tests, including if prematurely terminated and restarted. The remaining 25 marks will be available only to programs that are correct in all tests, and will be awarded based upon the efficiency of the code, i.e. how many mailboxes are used (the fewer the better) and how many fetch-execute cycles are used for each calculation (again the fewer the better).