Assignment #3 Fall 2011
SYSC-3006 Computer Organization
Department of Systems and Computer Engineering

Due Date: October 19th @ 7:00 PM (electronic submission)

You must submit the files identified below using the electronic Submit application. The submission process will be cancelled at the deadline. No assignments will be accepted via email or on disk.

Part I : Exercises in Control Flow and Basic Programming

1. **Arithmetic programming and Control Flow (assign31.asm, .exe):** Build a program that finds and displays the first 10 NEGATIVE ODD NUMBERS (i.e., -1, -3, -5, -7, -9...). You must do it in 2 different ways (both should be in the same program, one after the other):

   a) Using the MUL (and addition/subtraction) instructions.
   b) Using boolean and shift operations.

   The numbers must be printed in *decimal*. To print a character, you can use the following code:

```
; this uses DOS function 2 (of INT 21H support) to
;   output the character in the DL reg
        mov     ah, 2           ; DOS function 2
        int     21H             ; display character in DL
```

Hint: you have to convert the number to print into an ASCII code, move it to register DL, and then invoke the function `int 21H`, showed in the previous lines.

Note: read the Marking Scheme. It is highly recommended to work incrementally. Start by defining the code to compute the numbers. Then, work on the printing code. Each part will give you marks if correctly built. If you could not finish the printing routine, but you could check the right results using the debugger, leave a note as a comment in your source code, so the first part of the exercise will be marked.

2. **Arithmetic programming and Control Flow (assign32.asm, .exe):** Write a program to find and print the first 5 numbers of the *arithmetic progression.* The arguments of the progression should be defined as Global Variables in your program. Submit a version *with common difference 2* (3, 5, 7, 9...) http://en.wikipedia.org/wiki/Arithmetic_progression.

3. **Boolean logic (assign33.asm, .exe):** Write a program that counts how many 1s you have in register BX. Store the number in AL and display the number on screen (hint: you need to convert the number in AL into ASCII; to do that, add that number to ASCII ‘0’ before displaying). **You can only use boolean (AND, OR, NOT, XOR)**

or shift/rotate instruction for counting the 1s. Note: you can obtain partial marks if you don’t use Boolean operators.

4. **Basic Array Programming (assign34.asm, .exe).** Modify Exercise 1 and STORE THE RESULTS IN AN ARRAY. After, multiply each element of the array by 4. Display the contents of the array at the end of each step of the program (after the numbers have been generated and stored in the array, and after the numbers have been multiplied by 4).

5. **[Bonus] (assign35.asm, .exe)** Modify exercise 2, and store the first 6 elements you computed in an array. Then, print these 6 elements from the array, one by one. Then, compute the sum of all the elements in the array and display the result on screen.

Assignment 3 Marking Criteria (5 marks + 1 bonus total):

No marks will be given if the assignment is late or does not assemble. You may submit incomplete assignments if there are some working portions, but the submission must assemble and run for those portions AND you must clearly identify which parts of complete with comments. YOU LOSE ALL THE MARKS IF THE ROUTINE IS NOT PROPERLY DOCUMENTED.

Exercise 1: 1.25 marks total
0.5 marks: printing routine works properly
a) 0.25 + b) 0.5 marks: computing routines correct (using each of the 2 different strategies)

Exercise 2: 1 mark.

Exercise 3: 0.75 mark
0.5 marks: Values computed using Boolean logic operators only

Exercise 4: 1 mark total

Exercise 5: 1 mark total
0.25 marks: Data stored in Array correctly
0.25 marks: Data extracted from the Array (to print) correct
0.25 marks: Sum computed correctly
0.25 marks: displaying numbers correct