CSCIU210 Assembly Language Program 1
Due Nov 17th, 2017, Submit to Blackboard This assignment is worth 100 points

It is finally time to write your first assembly program. YEAH! This project is designed to strengthen your understanding of Assembly Language programming basics. The various items discussed in class will be used to complete this assignment. This assignment is to be **submitted to Blackboard by 11:59PM on Nov 17th**. This assignment should be completed individually. Keeping in mind that there are multiple ways to write the same program, therefore, no two programs should be the same. All work submitted should be professional, legible, and clearly show your programming solution for the problem. All code is subject to comparison among the other class members' submitted work, as well as may be run through a plagiarism detector.

WRITE YOUR OWN CODE!

Code that does not assemble/build into an *EXE file will receive a Zero. Your code must Build AND create an executable program. This is a bare minimum.

If you are working on an item/section(s) that you cannot get to work, then COMMENT it out, but leave it in your *.ASM code file so that I know you were at least trying to get the code to work for that item/section/requirement. Also, all issues and non-working requirements must be discussed in the Code Discussion comment entry.

Deliverables

1. ***.ASM file** containing your code **(85%)**. Titled yourinitials_alp1.asm
2. ***.EXE file** of your working program **(5%)**. Titled yourinitials_alp1.exe

   You will copy the Project.exe file to another location in which you should store your completed projects. You will then remain the Project.exe file to yourinitials_alp1.exe.
3. **Screenshot** of your running program's DOS console output **(5%)**
4. **Assignment Code Discussion comment** on Blackboard submission page (See Below)

   Code Discussion: In the Project Assignment comment section on Blackboard, (where you submit your assignment), include a brief paragraph discussing any issues that you encountered with the project and methods you used to overcome them. Discuss things you learned from the assignment and any confusion that you may still have on the material addressed in this assignment. **(5%)**
5. No hardcopy file required.

Skills Needed

- Basic understanding of Assembly Language programming
- Basic understanding of how to setup the Visual Studio environment
- Basic understanding of how to use the Visual Studio environment to open, modify, build and debug *.asm assembly programs.
- Basic understanding of the various components of an assembly language program, to include, but may not be limited to:

  .368
  .code
  .data
  .model flat, stdcall
  add
  call Crlf
  call DumpRegs
  call ReadInt
  call WaitMsg
  call WriteBin
  call WriteHex
  call WriteInt
  call WriteString
  Carriage Return
  Comments use
  dec
  Declaring DWORD variables which are uninitialized
  END
  ENDP
  exit
  Formatting protocols
  imul
  inc
  INCLUDE Irvine32.inc
  INCLUDE kernel32.lib
  Line Feed
  main PROC
  mov
  neg
  registers
  WriteString PROTO
  Writing a string of text to console (review inputloop.asm)

All the information you need for this project is available on Blackboard. Ensure to view the vast amounts of material there prior to searching the Internet. Review the *.asm examples for code references.

Program Hint 1: Write one section/item/component at a time, then Save, Build and Debug. If it works, then write another section/item/component. Then Save, Build and Debug. So on and so on.

Program Hint 2: Don't worry about errors in the Output window as long as your program runs properly and delivers the expected results. Some errors, such as a second incidence of .Model or that it cannot locate a file referenced in a library, do not necessarily interfere with your program operation.

Assembly Language Program Project 1 Instructions and Requirements:

1. **Header comment requirements**
    a. **Author** – Your Name
    b. **Course Title_Section** – CSCIU210_01 (or 02)
    c. **Creation Date** – date you write program
    d. **Program name** - such as "Assembly Language Program Project 1"
    e. **Revision** – should be 1, unless you are working on updates after creation
    f. **Program Description** – brief description of the program
2. **Program code comment requirements**
    a. Comments must be placed after each instruction line of code, preferably on the same line as the code, indicating what the line of code is doing. The comments should be brief, but contain enough detail that show you understand what the line of code is doing, as well as how that line of code affects any registers or flags.
    b. If you use outside class resources to complete a section of code to further your understanding of a specific directive, instruction, etc, please provide a comment with the URL link to the resource. You will not be penalized if you are using examples to learn from. However, do not copy other's code.
    c. Comments for the data section are not required, but you are welcome to use them.
3. **Program code requirements**
    a. Program should meet the requirements discussed in this document.
    b. Program should be neatly written and organized, such as was discussed in class. Use of formatting, such as indentation, comment alignment, standard use of naming conventions, standard use of case/capitalization, etc. There are programming protocols that you should use to make your code more legible by others. This is what is expected here. Your code may function, but if it isn't legible by others, then there will be issues.
    c. Submit softcopies as listed above under Deliverables to Blackboard.

Program Description

Create an assembly file *YourInitials_ALP1.asm*. For example TDM_ALP1.asm. The program should print the following information to the command console, as well as take user input and perform some basic arithmetic functions. Review the program output section for an example of what your finish output should look like. Numbers below correspond to line numbers in the command console output.

1. Blank line
2. Print your full name
3. Print assignment name
4. Blank line
5. Blank line
6. Prompt the user to enter a number and display the number on the same line.
7. Print statement confirming User entered number and their number
8. Print statement and hexadecimal equivalent of User entered number
9. Print statement and binary equivalent of User entered number
10. Blank line
11. Prompt the user to enter another number and display the number on the same line.
12. Print statement confirming User entered number and their number
13. Print statement and hexadecimal equivalent of second User entered number
14. Print statement and binary equivalent of the second User entered number
15. Blank line
16. Print statement and the sum of the two User entered numbers
17. Blank line
18. Print statement and the product of the two User entered numbers
19. Blank line
20. Print statement and User entered first number incremented by 2
21. Continue statement on second line so that User does not have to scroll.
22. Print statement and User entered second number decremented by 3
23. Continues statement on second line so that User does not have to scroll.
24. Blank line
25. Print statement and sum of new numbers
26. Blank line
27. Print statement and product of new numbers
28. Blank line
29. Prompt user to enter their favorite lucky number and display number on same line
30. Blank line
31. Statement that it is your favorite number too, include the number within statement.
32. Blank line
33. Statement about 2’s complement of lucky number being shown in EAX register.
34. Blank line
35. Dump of registers which include correct 2’s complement in EAX register
36. Blank line
37. Statement about showing off your Assembly Language skills
38. Blank line
39. Statement about the program ending successfully
40. Blank line
41. Prompt the user to “Press any key…”

CSCIU210 Assembly Language Program 1
Due Nov 17th, 2017, Submit to Blackboard This assignment is worth 100 points

Output console Example (output should look like the following, including spacing and output strings)

My name is *Your full name here*
Assembly Language Project 1

Please enter a signed integer: *User number*

The number you entered is: result
The hexadecimal equivalent is: result
The binary equivalent is: result

Please enter another signed integer: *User number*

The number you entered is: result
The hexadecimal equivalent is: result
The binary equivalent is: result

The sum of the two integers you entered is: result

The product of the two integers you entered is: result

If the first number you entered would have been 2 numbers greater,
It would have been: result

If the second number you entered would have been 3 numbers less,
It would have been: result

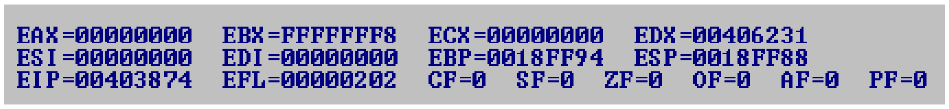
Had you entered these numbers, your sum would have been: result

Had you entered these numbers, your product would have been: result

Please enter your favorite lucky number: *User number*

Interesting, your favorite lucky number is result, so is mine!

The Two's Complement of your favorite lucky number is shown in the EAX register below.



```
EAX=00000000  EBX=FFFFFFF8  ECX=00000000  EDX=00406231
ESI=00000000  EDI=00000000  EBP=0018FF94  ESP=0018FF88
EIP=00403874  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0
```

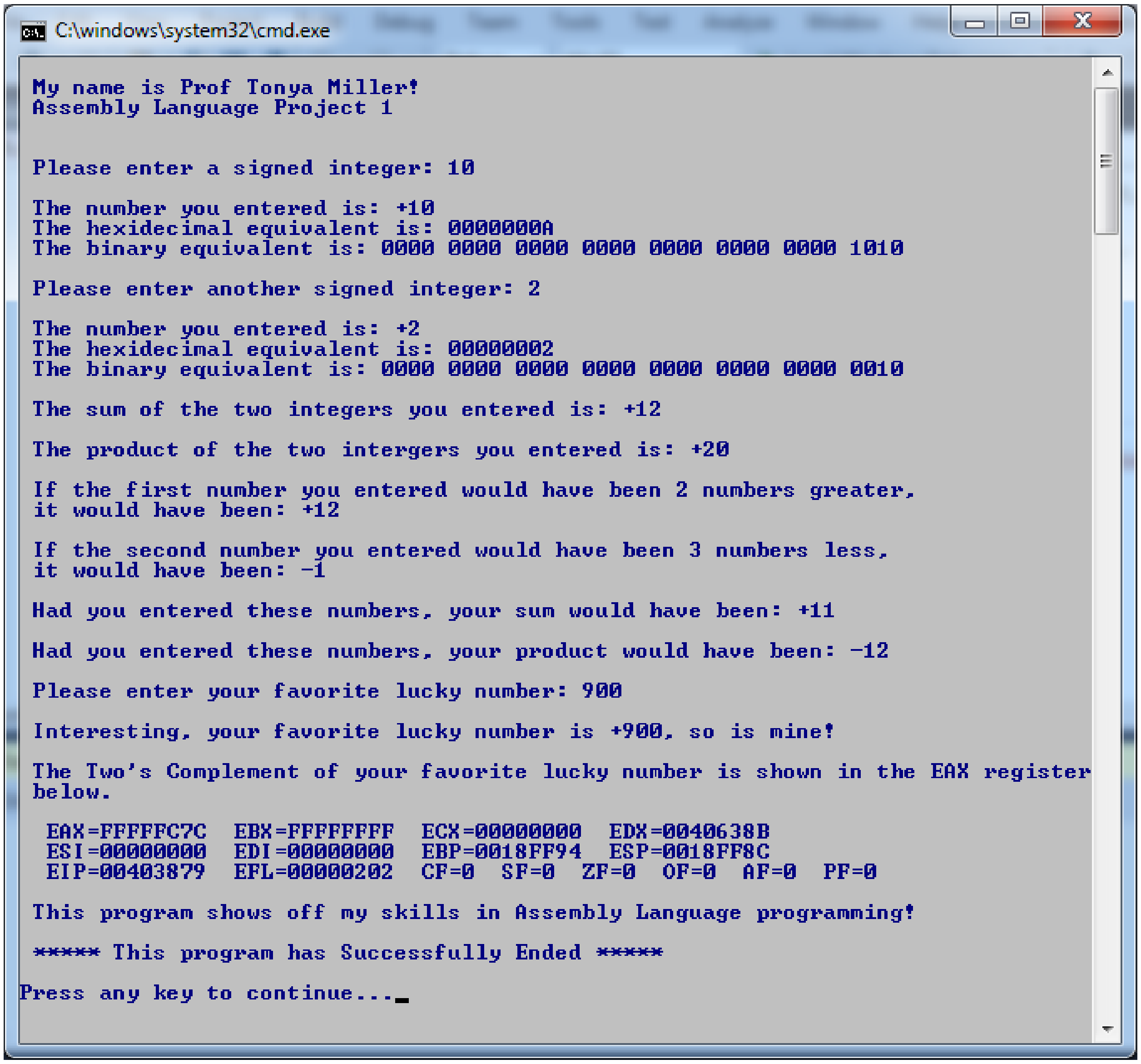
This program shows off my skills in Assembly Language programming!

***** This Program has Successfully Ended *****

Press any key to continue . . .

Note: Red text implies user input to the console terminal
Blue text implies results from program.

Screen shot of program with User entered values of 10, 2 and 900



```
My name is Prof Tonya Miller!
Assembly Language Project 1


Please enter a signed integer: 10

The number you entered is: +10
The hexidecimal equivalent is: 0000000A
The binary equivalent is: 0000 0000 0000 0000 0000 0000 0000 1010

Please enter another signed integer: 2

The number you entered is: +2
The hexidecimal equivalent is: 00000002
The binary equivalent is: 0000 0000 0000 0000 0000 0000 0000 0010

The sum of the two integers you entered is: +12

The product of the two intergers you entered is: +20

If the first number you entered would have been 2 numbers greater,
it would have been: +12

If the second number you entered would have been 3 numbers less,
it would have been: -1

Had you entered these numbers, your sum would have been: +11

Had you entered these numbers, your product would have been: -12

Please enter your favorite lucky number: 900

Interesting, your favorite lucky number is +900, so is mine!

The Two's Complement of your favorite lucky number is shown in the EAX register
below.

  EAX=FFFFFC7C  EBX=FFFFFFFF  ECX=00000000  EDX=0040638B
  ESI=00000000  EDI=00000000  EBP=0018FF94  ESP=0018FF8C
  EIP=00403879  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0

This program shows off my skills in Assembly Language programming!

***** This program has Successfully Ended *****

Press any key to continue..._
```