

COSC2406-F18 – Assembly Language Programming

Assignment 3

Due: Oct 21st 11th, 2018 by 11:55pm

In order to complete this assignment, you need to setup MASM (and Visual Studio) on your home computer or laptop if this is where you intend to do your assignments. If you plan to use the lab computers, go to NW201 and make sure the MASM environment is working.

If you have any issue with the questions or unclear about the requirements, please email me immediately.

General Requirements:

- a) For each of the following questions, you must first construct a Java program which is limited to using:
 - a. addition and/or subtraction
 - b. contains just a main method
 - c. has only global static variables (no local variables)
 - d. can use loops
 - e. can use single dimension arrays (they must be static data members of the class) and the length property of the array

You cannot use:

- a. Local variables**
 - b. IF/SWITCH statements**
 - c. Multiplication, division, modulus**
 - d. Math library functions**
- b) You must fully comment your assembly programs. The full code the Java Program is to be included in the comments of the header. Additionally, you can use some of the instructions to comment your assembly code.
- c) The IRVINE library will be used in all the programs. No other libraries can be used.
- d) *Only assembly operation instructions covered to the end of Chapter 4 + the library functions described in chapter 5 (Section 5.4) can be used.***

Questions: Produce both Java and Assembly programs for all

- A) [20 marks] To collect an unsigned integer value from a user, you need to first ask a question and then collect the response. To ask a question in assembly, you can use the **WriteString** library function to print a string by placing the offset of the string in the EDI register and then call the function. To collect the unsigned integer value, use the **ReadDec** library function. The value is returned in the eax register. To collect a signed integer, the **ReadInt** library function is used instead.

COSC2406-F18 – Assembly Language Programming

Write a program which prompts the user to enter five numbers and store the numbers into five double word variables, P, Q, R, S, & T, in the data segment. Each prompt must be different and specific. P, R and T are unsigned values whereas Q and S are signed values. Once collected, use the five values to solve and report the results for each of the following equations using only the EBX and EAX registers:

$$(P + R) - (Q + S) + T$$

$$T + (P - R) - (S + Q)$$

$$(S + Q) + (T - P) - R$$

To print a signed integer number, you would make sure the number to print is in the EAX register and then call the *WriteInt* library function. Your output should look something like:

Enter a unsigned value for P: 14

Enter a signed value for Q: -8

...

$$(P + R) - (Q + S) + T = +453$$

$$T + (P - R) - (S + Q) = -1836$$

$$(S + Q) + (T - P) - R = +8634$$

- B) [15] Write a program which asks the user for their average driving speed along with the minimum and maximum hours they plan to drive. Using these numbers, create a table which shows how far the user will travel for each of the hours between the minimum to the maximum. The output would look something like this:

Please enter your average speed (km/hour): 55

Please enter the minimum number of hours you intend to drive: 6

Please enter the maximum number of hours you intend to drive: 10

Hours	Distance Driven
6	330
7	385
8	440
9	495
10	550

COSC2406-F18 – Assembly Language Programming

- C) [15 marks] Write an assembly program that uses a loop to calculate the first thirty values of a number sequence described by the formula: $F(0) = 0$, $F(1) = 2$, $F(2) = 1$, $F(3) = 4$, $F(4) = 2$, and then $F(n) = 2 * F(n-1) + F(n-5)$ for $n \geq 5$. Your program needs to print the values as a comma separated – the first four values are simply printed and the remaining values are calculated using a loop.

Place each value in the EAX register and display it using the **WriteDec** library function. Use the **WriteString** library function to print a string **“, ”,0** which is a comma, a space and a null terminator and the **CrLf** function to print a newline character when you are finished (see Chapter 5.4.3, pages 158-70). The output would be something like:

```
0, 2, 1, 4, 2, 4, 10, 21, 43, 88, ...
```

- D) [10 marks] Create two arrays – one with 10 WORD values and the second with 10 DWORD values – both initialized to undefined values. In the program code, initialize the value of the WORD array to have values from 10, 20, 30... all the way to 100. In a second loop, copy the values from the WORD array to the DWORD array. Finally, using a third loop, display the values in the DWORD array with a comma and a space between each. You may find the WriteChar function useful.

- E) [10 marks] Write a program that uses a loop and indirect addressing to copy a string from a source array of bytes to a target array of bytes in reverse order. Print both the original string (source) and the copied string (target) to show that the program works. The data section will include:

```
source BYTE "This is the original string", 0
target  BYTE SIFEOF source DUP ('')
```

The final output should be something like this:

```
This is the original string
gnirts lanigiro eht si sihT
```

Submission, demonstrating, and grading:

For each of the programs:

- (i) Include the Java program as a header comment in the code.
- (ii) The Java program can ALSO be used as the comments for the assembly code along with other descriptions as required.
- (iii) The header must include your name, assignment number, question number, and program description.

Upload all of your **.asm** files for each of the programs to CMS.

COSC2406-F18 – Assembly Language Programming

All programs must be fully commented using the two-column approach discussed in class and shown throughout the textbook in the code examples. ***Failure to comment the code properly or to format like the code in the textbook will result in a penalty of up to 25%.***

Demonstrate and explain your working programs to the TA by Thursday Oct 18th. The demonstration and explanation of your code will represent 20% of the assignment grade – the other 80% of the grade will come from grading the code and how well the programs meet the specification and requirements of the question.

IMPORTANT:

- a) ***If you do not demonstrate your programs to the TA by the end of the demonstration date, you will receive a grade of zero.***
- b) ***If you are unable to explain most or all of your code and the logic of your programs, then the maximum grade possible will be 50%.***