



**Assignment # 03**  
**Microprocessor & Assembly Language**  
**BS (CS)**  
**Fall semester 2018**

- Q.1** Using the value  $-35$ , write it as an integer literal in decimal, hexadecimal, octal, and binary formats that are consistent with MASM syntax.
- Q.2** Create a single integer expression that uses all the operators. Calculate the value of the expression.
- Q.3** Write the real number  $-6.2 \times 10^4$  as a real number literal using MASM syntax.
- Q.4** Discuss the following MASM directives:
- |         |       |        |        |       |
|---------|-------|--------|--------|-------|
| INCLUDE | .386  | .MODEL | .STACK | PROTO |
| .DATA   | .CODE | PROC   | ENDP   | END   |
- Q.5** Which statement halts the assembly language program?
- Q.6** What type of argument must be passed to the ExitProcess procedure?
- Q.7** What is a calling convention, and how is it used in assembly language declarations?
- Q.8** What types of files are produced by the assembler and linker?
- Q.9** Which operating system component reads and executes programs?
- Q.10** Create an uninitialized data declaration for a 08-bit, 16-bit, and 32-bit unsigned and signed integers.
- Q.11** Create a data definition for a doubleword that stored it in memory in little endian format.
- Q.12** Show the order of individual bytes in memory (lowest to highest) for the following doubleword variable:
- ```
val1 DWORD 87654321h
```
- Q.13** Find out if you can declare a variable of type DWORD and assign it a negative value.
- Q.14** Declare an array of 120 uninitialized unsigned doubleword values.
- Q.15** Declare an array of byte and initialize it to the first 5 letters of the alphabet.
- Q.16** Declare an unsigned 16-bit integer variable named **wArray** that uses three initializers.
- Q.17** Declare a string variable containing the name of your favorite color. Initialize it as a null-terminated string.
- Q.18** Why might you use a symbolic constant rather than an integer literal in your code?
- Q.19** Write a statement that causes the assembler to calculate the number of bytes in the following array, and assign the value to a symbolic constant named **ArraySize**:
- ```
myArray WORD 20 DUP(?)
```
- Q.20** Show how to calculate the number of elements in the following array, and assign the value to a symbolic constant named **ArraySize**:
- ```
myArray DWORD 30 DUP(?)
```
- Q.21** Write a program that defines symbolic constants for all seven days of the week. Create an array variable that uses the symbols as initializers.

- Q.22** Write a program that defines symbolic names for several string literals (characters between quotes). Use each symbolic name in a variable definition.
- Q.23** Differentiate between *equal-sign* directive and *EQU* directive.
- Q.24** Give examples of three different instruction mnemonics having zero, one, and two operands.
- Q.25** How is a source file different from a listing file?
- Q.26** Write a program that contains two instructions: (1) add the number 5 to the EAX register, and (2) add 5 to the EDX register. Generate a listing file and examine the machine code generated by the assembler. What differences, if any, did you find between the two instructions?
- Q.27** How are data labels and code labels different?
- Q.28** Name the four basic parts of an assembly language instruction.
- Q.29** Show an example of a block comment.
- Q.30** Why is it not a good idea to use numeric addresses when writing instructions that access variables?
- Q.31** Find out, by trial and error, if a program can have multiple code and data segments.
- Q.32** Write a program that calculates the following expression, using registers:

$$A = (A + B) - (C + D)$$

Assign integer values to the EAX, EBX, ECX, and EDX registers.