Daniel Martinez

CIS 11-34595

Assignment 3

**Section 3.1.11**

1. -35d, DDh, 335o, 11011101b
2. No it must be preceded by a 0 if the first character is a letter.
3. No they are the same precedence.
4. (2 + 3) \* ( 9 / ( 8 – 5 )) = 5 \* (9/3) = 15

**Section 3.2.4**

4. EAX

1. The “END main” command.

**Section 3.3.3**

1. Object files.
2. True, kind of, except it doesn’t extract the files it copies them and it doesn’t insert them into the executable file it combines them with the object file then creates an executable file.
3. True.
4. The loader.

**Section 3.4.13**

1. label SWORD ?
2. label BYTE ?
3. label SBYTE ?
4. label QWORD ?
5. SDWORD

**Section 3.5.5**

3. myArray WORD 20 DUP(?)

ArraySize = ($ - myArray)

**Section 3.9.1**

4. Because it still gets converted to machine language before executing.

5. Little endian order basically means that our processor stores and retrieves items starting with the “little end” or the end on the right because those positions in a number hold the ‘littlest’ value. Big endian, on the other hand, stores from left to right. I found that big endian comes from people wanting to make the computer store things similar to the way we read them. Little endian, comes from it being easier for the computer to store something that way as storing from right to left makes it so the computer does not have to keep moving everything back one space while it stores the data.

6. That way if the parameters for that variable happen to change you can simply change the variable value and all variables of that definition will take on the new data.

25. SDWORD

**Section 3.9.2**

4. You can however the computer will consider it to just be a higher positive value because it would assume the number is not in two’s compliment form.

7. myArray DWORD 120 (?)

13. BYTE 500 DUP (“TEST”)

**Section 3.10**

1.

Mov eax, 5

Mov ebx, 6

Mov ecx, 4

Mov edx, 2

Add eax, ebx

Add ecx, edx

Sub eax, ecx