Brian Nguyen

Assignment 3

- [ ] 3.1.11 Section Review, Questions 1, 2, 3, 4

- [ ] 3.2.4 Section Review, Questions 4, 5

- [ ] 3.3.3 Section Review, Questions 1, 2, 3, 4

- [ ] 3.4.13 Section Review, Questions 1, 2, 3, 4, 5

- [ ] 3.5.5 Section Review Question 3

- [ ] 3.9.1 Short Answer, Questions 4, 5, 6, 25

- [ ] 3.9.2 Algorithm Workbench, Questions 4, 7, 13

- [ ] 3.10 Programming Exercises, Pick any 1 exercise

- [ ] Upload your assignment to GitHub

- [ ] Email james.wilson@rccd.edu with the url to the source code on GitHub

**3.1.11 Section Review**

1. Decimal: -35d Binary: -100011b Hexadecimal: -23h Octal: -43o

2. Yes

3. No

4. (+3 - 2) / 2 \* 2 MOD 7 = 2

**3.2.4 Section Review, Questions**

4. Register EAX hold the sum.

5. the “END main” statement halts the program

**3.3.3 Section Review, Questions 1, 2, 3, 4**

1. The assembler produces object files.

2. True

3. True

4. The operating system loader reads and executes programs.

**3.4.13 Section Review**

1. SWORD ?

2. BYTE ?

3. SBYTE ?

4. QWORD ?

5. SDWORD can hold a 32 bit singed integer.

**3.5.5 Section Review**

3. myArray WORD 20 DUP(?)

ArraySize = ($ - myArray) / 2

**3.9.1 Short Answer**

4. The assembler is sort-of the compiler which translates code in assembly to machine code. “Assembler language” would not be correct as it is referring to the translator as opposed to “Assembly language”.

5. little-endian is where the least significant byte of a value is stored in the first memory address allocated for the data then onwards to the more significant bytes which are stored in the next memory addresses. Big-endian is opposite where the most significant byte is stored at the first and then the rest. The terms come from a story in the 1700s in which a war erupts due to argument over if an egg should be cracked at the big-end or little-end.

6. You might use a symbolic constant instead of a integer literal in your code because if the value must be changed latter on you can simply set value of the symbolic constant to another as opposed to having to change every single value of the integer literal wherever it appears.

25. SDWORD

**3.9.2 Algorithm Workbench**

4. The assembler allows you to assign DWORD a negative value. This shows that the type checking is not very restrictive at all which is typical of assembly having less strict rules for what can be done.

7. myArray DWORD 120 DUP(?)

13. bArray BYTE 20 DUP(0)

**3.10 Programming Exercises**

Programming Exercise #1 (In Visual Studios Project Folder)

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO,dwExitCode:DWORD

.code

main PROC

mov eax,5 ;A

mov ebx,6 ;B

mov ecx,1 ;C

mov edx,2 ;D

add eax,ebx ;(A+B)

add ecx,edx ;(C+D)

sub eax,ecx ;A = (A+B) - (C+D)

INVOKE ExitProcess,0

main ENDP

END main