

Assignment 2

"I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work."

Name	Ravi Trivedi
Student Number	105197609
UWinID	trive11w@uwindsor.ca

Section 1

Question 1:

1. List four different instruction mnemonics. (2 points)

➤	Mov
➤	Add
➤	Sub
➤	Call

2. Name the four basic parts of an assembly language instruction. (2 points)

➤	Mnemonic
➤	Operands
➤	Label
➤	Comment

3. A command that is recognized and executed by the assembler while the source code is being assembled is a(n) _____. (1 point)

a. Answer> Directive

4. Use a TEXTEQU expression to redefine "PROC" as "PROCEDURE." (2 points)

a. Answer> PROCEDURE TEXTEQU <PROC>

5. Show a brief example of a block comment. (1 point)

a. Answer>

COMMENT & Some Comment Some other Information Some more Details &

6. Show the order of individual bytes in memory (lowest to highest) for the following doubleword variable: val DWORD 78563412h (2 points)

a. Answer> 12h, 34h, 56h, 78h

Question 2:

1. What is the main difference between data labels and code labels? (4 points)

a. Code labels are always in code segment and is generally utilized for loop control statements as per Data labels are in data segments and are known as variable offsets.

2. What is the main difference between source file and listing file? (4 points)

a. A source file is an .asm file that comprise the assembly code written inside and then send that to the assembler as an input.

b. A listing file is an .lst file that comprise the instruction offset addresses, machine code. Listing file is generated by assembler.

3. What is the main difference between big endian and little endian? Also, look up the origins of this term on the Internet. (4 points)

a. The main difference between big endian and little endian is how they both places the least significant bit.

b. Little endian places LSB in position 0. On the contrary, big endian places MSB in position 0.

4. Explain using example the process of calculating the sizes of

a. Arrays

b. Strings

c. Word

d. Dword

➤ **Arrays:**

When using arrays, the '\$' operator returns the current location counter. In order to get the size of the array, we simply subtract the list from current location counter.

EX:

arr BYTE 1,2,3,4 ; Array defined.

arrSize = (\$ - arr) ; Calculating the size of arr

➤ **Strings:**

We can also use the same technique as above to find the length(size) of the string.

EX:

str BYTE "This is a string." ; String defined.

strSize = (\$ - str) ; Calculating the size of str

➤ **Word:**

The technique is still the same, but for array containing the values other than BYTE, In this example WORD, We need to divide the calculation by the array elements size, In this case 2.

EX:

arrWord WORD = 100h, 101h, 102h, 103h ; Word array defined.

arrWSize = (\$ - arrWord) / 2 ; dividing by two because arr contains WORD elements

The technique is still the same, but for array containing the values other than BYTE, In this example DWORD, We need to divide the calculation by the array elements size, In this case 4.

EX:

```
arrDWord DWORD = 100h, 101h, 102h, 103h
```

```
; DWord array defined.
```

```
arrDWSize = ($ - arrWord) / 4
```

; dividing by four because arr

; contains DWORD elements