Assembly Language

Assignment 7

This Assignment will help you to save time understanding a lot of important points in this course. Give this assignment a great attention along with previous assignments.

- Individual task.
- **Due:** Next section **(**one night before if you submit by email**)**

Q 1) (5 Points) Use the following data definitions to solve the next exercises:

```
.data
myBytes    BYTE     30h,70h,0A0h,40h,60h
myWords    WORD  3 DUP(?),4000h
myString   BYTE     "hello!!"
```

a) What will be the value of EAX after each of the following instructions execute?

```
mov  eax, TYPE          myBytes          ; a.
mov  eax, LENGTHOF  myBytes          ; b.
mov  eax, SIZEOF        myBytes          ; c.
mov  eax, TYPE          myWords          ; d.
mov  eax, LENGTHOF myWords          ; e.
mov  eax, SIZEOF        myWords          ; f.
mov  eax, SIZEOF        myString         ; g.
```

b) Write a single instruction that moves the first two bytes in myBytes to the DX register.

c) Write an instruction that moves the second byte in myWords to the AL register.

d) Write an instruction that moves the first four bytes in myBytes to the EAX register.

e) Insert a LABEL directive in the given data that permits myWords to be moved directly to a 32-bit register.

Q 2) (5 Points) Trace the following program then answer the questions:

```
1.  INCLUDE Irvine32.inc
2.  .data
3.
4.  byteVal       byte 1,2,3,4
5.  wordVal       word 1000h,2000h,3000h,4000h
6.  dwordVal      dword 12345678h, 87654321h
7.  aString       byte "ABCDEFG",0
8.  pntr          dword wordVal
9.
10. .code
11. main PROC
12.
13.       ; The following 4 lines to print 32 byte of the .data segmant starting from byteVal
14.       mov esi, offset byteVal
15.       mov ecx, 32
16.       mov ebx, type byteVal
17.       call   dumpmem
18.
19.       mov eax,0           ;clear eax with zeros
20.       mov ebx,0           ;clear ebx with zeros
21.
22.                           ;no need to clear ecx, edi, esi, as it come with zeros by defult
23.       mov esi, offset byteVal
24.       mov ebx, offset byteVal
25.       inc byte ptr [esi]
26.       inc esi
27.
28.       mov ah, byte ptr wordVal
29.       mov bx, word ptr byteVal+4
30.       mov cx, word ptr Astring
31.       mov dx, word ptr pntr;
32.
33.       call dumpregs
34.
35.       exit
36. main ENDP
37. END main
```

A. After tracing this program manually, write down the contents of memory for the ".data segment" of this program, and the value of the following registers:

   AH= BX= CX= DX= ESI=.

B. Using MASM test this program and compare the results with your answer for point A.

C. Remove "byte ptr" from line 25, run the program and indicate whether MASM gives error or not.

D. Change line 25 into (inc word ptr [esi]), run the program and indicate whether MASM gives error or not.

E. Change line 26 into (inc byte ptr esi), run the program and indicate whether MASM gives error or not.

F. Change line 25 into (inc byte ptr [ebx]), run the program and indicate whether MASM gives error or not.

G. Change line 28 into (mov [ebx], [esi]), run the program and indicate wheter MASM gives error or not.

Q 3) (5 Points) Trace the following program, and then answer the questions:

```
1.  INCLUDE Irvine32.inc
2.  .data
3.
4.  arrayA word   01h, 02h,
5.                03h, 04h
6.  aSize  byte  ($-arrayA)
7.
8.  arrayB word   01h, 02h
9.                word 03h, 04h
10.
11. .code
12. main PROC
13.
14.       mov eax, 0          ;clear eax with zeros,
15.       mov al, aSize
16.       mov ah, lengthof arrayA
17.
18.       mov ebx,0           ; clear ebx with zeros
19.       mov bl, sizeof arrayA
20.
21.                           ; no need to clear ecx,esi,edi  as it come with zeros by defult
22.       add cx, [arrayA+esi]
23.       add esi, 2
24.       add cx, arrayA[esi]
25.       mov edi,esi
26.       add cx, arrayA[edi*type arrayA]
27.       add cx, arrayA[edi+(2*type arrayA)]
28.
29.       mov edx,0           ; clear edx with zeros
30.       mov dl, lengthof arrayB
31.       mov dh, byte ptr arrayB+4
32.
33.       call dumpregs
34.       exit
35. main ENDP
36.
37. END main
```

A. After tracing this program manually, what is the value of:

   AL= AH= BL= BH CX= DL= DH= ESI= EDI=

B. Using MASM test this program and compare the results with your answer for point A.

C. Change line 6 with (aSize word ($-arrayA)), then run the program and indicate whether MASM gives error or not.

D. Change line 22 with (add cx, arrayA+esi), then run the program and indicate whether MASM gives error or not.

E. Change line 30 with (mov dl, sizeof arrayB), then run the program and indicate the value of AL.

F. Remove “byte ptr” from line 31, run the program and indicate wheter MASM gives error or not.