

Microprocessor & Assembly Language

Program: BS (CS)

Course Code: CSC-304

EDP Codes: 102007054

Instructor: Muhammad Amin

Examination: Final-Term

Semester: Summer 2020

Duration: 4 Hours

Date: Sep. 26, 2020

Time: 9:00 am

| Question No. | Q.1 | Q.2 | Q.3 | Q.4 | Q.5 | Q.6 | Q.7 | |
|--------------|-----|-----|-----|-----------|-----|-------|------------|----|
| Total Marks | 16 | 12 | 8 | 7+3+3= 13 | 6 | 3x3=9 | 3+3+4+6=16 | 80 |

Note: Attempt all questions.

Q.1 Give answers to each of the following:

- Differentiate between equal-sign directive and EQU directive.
- Differentiate between data label and code label.
- What is a calling convention, and how is it used in assembly language declarations?
- Explain different instruction mnemonics having zero, one, and two operands.
- Using the value -1, write it as an integer literal in decimal, hexadecimal, octal, and binary formats that are consistent with MASM syntax.
- Write the real number -6.2×10^6 as a real number literal using MASM syntax.
- Show the order of individual bytes in memory (lowest to highest) for the following doubleword variable: `val1 DWORD 12345678h`
- 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 60 DUP(?)`

Q.2 Use the following variable definitions for the coming question:

```
.data
wVal1      WORD 3000h
wVal2      WORD 7000h
listB      BYTE 50h, 40h, 30h, 20h, 10h
listW      WORD 3000h, 2000h, 1000h
listD      DWORD 30000000h, 20000000h, 10000000h
```

.code

main PROC

What will be the value of the destination operand after each of the following instructions execute in sequence?

```
mov bx, 0ABCDh
movzx eax, bx ; (a) EAX =
```

```

mov bx, 0DCBAh
movsx eax, bx           ; (b) EAX =
mov ax, wVal1          ; (c) AX  =
xchg ax, wVal2         ; (d) AX  = , (e) val2 =
mov wVal1, ax          ; (f) val1 =
mov al, listB          ; (g) AL  =
mov al, [listB+4]      ; (h) AL  =
mov ax, listW          ; (i) AX  =
mov ax, [listW+4]      ; (j) AX  =
mov eax, listD         ; (k) EAX =
mov eax, [listD+8]     ; (l) EAX =

```

Q.3 Write down the values of the Carry, Sign, Zero, and Overflow flags after each instruction has executed:

```

mov ax, 7FF0h
add al, 10h           ; (a) CF = SF = ZF = OF =
add ah, 1             ; (b) CF = SF = ZF = OF =
add ax, 2             ; (c) CF = SF = ZF = OF =
mov al, 1
sub al, 2             ; (d) CF = SF = ZF = OF =

```

Q.4 Use the following data definitions for the coming question:

```

.data
listB      BYTE 60h, 50h, 40h, 30h, 20h, 10h
listW      WORD 4 DUP(?), 1000h
string1    BYTE "Assembly Language", 0

```

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

```

mov eax, TYPE listB      ; (a) EAX =
mov eax, LENGTHOF listB  ; (b) EAX =
mov eax, SIZEOF listB    ; (c) EAX =
mov eax, TYPE listW      ; (d) EAX =
mov eax, LENGTHOF listW  ; (e) EAX =
mov eax, SIZEOF listW    ; (f) EAX =
mov eax, SIZEOF string1  ; (g) EAX =

```

(ii) Write an instruction that moves all four bytes in listB to the EAX register.

(iii) Insert a LABEL directive in the given data that permits listB to be moved directly to EAX register.

Q.5 Use the following data definitions for coming question:

```
listB      BYTE 10h, 20h, 30h, 40h
listW      WORD 8Ah, 3Bh, 72h, 44h, 66h
listD      DWORD 1, 2, 3, 4, 5
pointer1   DWORD listD
```

What will be the value of the destination operand after each of the following instructions execute in sequence?

```
mov esi, OFFSET listB
mov al, [esi]                ; (a) AL =
mov al, [esi+3]             ; (b) AL =
mov esi, OFFSET listW + 2
mov ax, [esi]               ; (c) AX =
mov edi, 8
mov edx, [listD + edi]      ; (d) EDX =
mov edx, listD [edi]        ; (e) EDX =
mov ebx, pointer1
mov eax, [ebx+4]            ; (f) EAX =
```

Q.6 Implement the following pseudocode in assembly language:

```
(i)   if( var1 <= var2 )
        var3 = 15;
    else
    {
        var3 = 10;
        var4 = 30;
    }
(ii)  if ( val1 > ecx ) AND ( ecx > edx ) then
        A = 12
    else
        B = 6;
(iii) while( ebx < eax)
        ebx = ebx + 1;
```

Q.7 (i) What will be the final value of EAX in this example?

```
mov eax, 0
mov ecx, 10
L1:  mov eax, 3
     mov ecx, 5
L2:  add eax, 5
     loop L2
     loop L1
```

(ii) Write a program that calculates the following expression, using registers:

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

(iii) Write a program that uses a loop to copy all the elements from an unsigned Word array into an unsigned doubleword array.

(iv) Write a program that displays a string in all possible combinations of foreground and background colors (16 x 16 =256). The colors are numbered from 0 to 15, so you can use a nested loop to generate all possible combinations. Also use a delay of 1s in each foreground color change.

******End of Exam******