Sir Syed University of Engineering & Technology
ANSWER SCRIPT

Date:	June 12,2021
Roll Number:	CS19-037
Section:	A
Name:	Munib ul Hassan
Course Name:	CS-330: Microprocessor & Assembly Language
Degree Program:	BSCS
Total number of pages being submitted:	8

My Birth Year is 2000

D_1 = 2000

My Date of Birth is 02-10-2000

D_2 = 0 + 2 + 1 + 0 + 2 + 0 + 0 + 0 = 5

My Month of birth is 10

D_3 = 10

My Roll no is 37

D_4 = 37

My Date of Birth is 02-10-2000

D_5 = 2

ANSWER 01:

a. $D_1 = 2000_{16}$

D_1 into decimal

$= 2 * 16^3 + 0 * 16^2 + 0 * 16^1 + 0 * 16^0$

$= 2 * 4096 + 0 + 0 + 0$

$= 8192 + 0 + 0 + 0$

$D_1 = 8192_{10}$

$D_2 = 5_{10}$

D_2 into Hexadecimal

It is same as in decimal

$D_2 = 5_{16}$

b. Find the 2's compliment of D_3 and D_5 (Hexaecimal)

$D_3 = 10_{10}$ For convert into 2's compliment 1. Substract from FFFFFFFF 2. Add 1 FFFFFFFF -0000000A FFFFFFF5 + 1 FFFFFFF6 2's Compliment of D_3 is $FFFFFFF6_{16}$	$D_5 = 2_{16}$ For convert into 2's compliment 1. Substract from FFFFFFFF 2. Add 1 FFFFFFFF -00000002 FFFFFFFD + 1 FFFFFFFE 2's Compliment of D_5 is $FFFFFFFE_{16}$

c. Peform the following operaions on the given Data:

i. D_4 / D_2 ii. $D_3 - D_4$

D_4 / D_2 $\frac{37}{5} = 7.4$ $\frac{D_4}{D_2} = 7.4_{10}$	$D_3 - D_4 = 10 - 37 = -27$ **$D_3 - D_4 = -27$**

d.

To Represent $-D_4$ To get the two's complement negative notation of an integer, you write out the number in binary. You then invert the digits, and add one to the result. $D_4 = 37_{10}$ $= 00100101_2$ Invert the digits $= 11011010_2$ Add 1 11011010_2 $+ 1$ 11011011_2 **$D_4 = 11011010_2$** **$-D_4 = 11011011_2$**	To Represent $-D_2$ (Hex) To get the two's complement negative notation of an integer, you write out the number in binary. You then invert the digits, and add one to the result, then convert this into hexadecimal. $D_2 = 5_{10}$ $= 00000101_2$ Invert the digits $= 11111010_2$ Add 1 11111010_2 $+ 1$ 11111011_2 Convert this in to hexadecimal $1111\ 1011_2$ FB_{16} **$D2 = 5_{16}$** **$-D2 = FB_{16}$**

e. $D_1 = 2000$

In UnPacked BCD	In Packed BCD
= 00000010 00000000 00000000 00000000 Total no of bytes for Unpacked BCD: 4 bytes	= 0010 0000 0000 0000 Total no of bytes for Packed BCD: 2 bytes

ANSWER 02:

a. Physical address = $(D_1 * 10)h$

As we know that

Physical address = Segment Address * 10h + offset Address

By comparing we have:

Segment address : 2000

Offset Address: 0000

Logical address: segment address: offsetAddress

Logical address = 2000 : 0000

b. Physical memory location = D_4 = 37h = 37 * 1000= 37000h
DS = D_1 = 2000
Lower range for DS = (2000*10) + 0000 = 20000
Upper range for DS = (2000*10) + FFFF= 2FFFF

37000 is greater than range.
The requires value for DS is:

Offset = D_2 = 5

Physical address = Segment Address * 10h + offset Address

37000 = (DS * 10) + 5
37000 – 5 = DS * 10
36995 = DS * 10
DS = 3699.5h

ANSWER 03:

a. $D_1 = 2000_{10}$

Convert this into Hexadecimal

16 | 2000
16 | 125 → 0
| 7 → D

$D_1 = 7D0_{16}$

For no of count in D_1 we mst convert it into Binary

$D_1 = 2000_{10} = 11111010000_2$

No of ones in D_1 = is 6

6 in BCD = 00000110_2

CODE:

```
.Model Small
.Stack 100h
.Data
        DATA1 DW 7D0h
        COUNT DB ?
.Code
MAIN PROC
        MOV AX,@DATA      ;to initialize DS
        MOV DS,AX
        SUB AL, AL
        MOV DL, 16
        MOV BX, DATA1
        AGAIN:
                ROL BX, 1
                JNC NEXT
                ADD AL, 1
                DAA
        NEXT:
                DEC DL
                JNZ AGAIN
                MOV COUNT, AL
        Mov AH, 4CH
        Int 21h
MAIN ENDP
END MAIN
```

b. CODE

```
        .Title Q3b
        .Model Small
        .Stack 100h
        .Data
                value DW 2000h,5h,10h,37h,2h
                Result DB ?
        .Code
        MAIN PROC
```

```
        MOV AX,@DATA      ;to initialize DS
        MOV DS,AX
        MOV CX,4
        MOV BX, OFFSET value
        MOV AL, [BX]
        LBACK:
            CMP AL, [BX+1]
            JC SW=AP
        SBACK:
            INC BX
            LOOP LBACK
            JMP TER
        SWAP:
            MOV AL, [BX+1]
            JMP SBACK
        TER:
            MOV Result, AL
            MOV AH,4CH
            INT 21H
    MAIN ENDP
    END MAIN
```

ANSWER 04:

a. CODE:

```
.Title Q5a
.Model Small
.Stack 100h
.Data
    D2 DB 5
    D4 DB 37
.Code
MAIN PROC
    MOV AX,@DATA      ;to initialize DS
    MOV DS,AX
    MOV AL,D2
    MOV BL,D4
    ADD AL,BL
    MOV AH,4CH
    INT 21H
MAIN ENDP
END MAIN
```

37 00100101
+5 +101
42 00101010
OF = 0 CF = 0 PF = 0 AF = 0 SF = 0 ZF = 0

b. CODE:

```
.Title Q4B
.Model Small
.Stack 100h
.Data
        ASCII  DB '2000,5,10,37,2'
        BCD DB 5 DUP(?)
        STRING DW 'INCOREECT DATA'
.Code
MAIN PROC
        MOV AX,@DATA      ;to initialize DS
        MOV DS,AX
        LEA SI, ASCII
        LEA DI, BCD
        MOV CX,5

LOOP:
        MOV AX,[SI]+CX
        SUB  AX, 3030H
        ROL AH , CX
        ADD  AH ,AL
        MOV [DI]+CX, AH
        DEC CX
        JN ALERT
        JNZ LOOP
ALERT:
        LEA DX,STRING
        MOV AH,09H
        INT 21H
MOV AH,4CH
INT 21H

MAIN ENDP
END MAIN
```

ANSWER 05:

a. Name : "Munib"

CODE:

```
.Title Q5a
```

```
.Model Small
.Stack 100h
.Data
        HEX DB ‘6d, 75, 6e, 69, 62’
        ASCII DB ?
.Code
MAIN PROC
        MOV AX,@DATA      ;to initialize DS
        MOV DS,AX

        MOV AL,OFFSET HEX
        AND AL,0FH
        CMP AL,09H
        JBE DOWN
        ADD AL,07H
DOWN:
        ADD AL,30H
        MOV CL,05H
        ADD AH,0F0H
        ROL AH,CL
        JBE UP
        ADD AH,07H
UP:
        ADD AH,30H
        MOV ASCII,AX

        MOV AH,4CH
        INT 21H
MAIN ENDP
END MAIN
```

b. CODE:

```
.Title Q5b
.Model Small
.Stack 100h
.Data
        String DB ‘ANdul Ghani’
.Code
MAIN PROC
        MOV AX,@DATA      ;to initialize DS
        MOV DS,AX
        MOV ES, AX
```

```
        LEA DI, Sting
        MOV CX, 11
        MOV AL,'b'
        REPNE SCASB
        JNE OVER
        DEC DI
        MOV BYTE PTR [DI],'B'
OVER:
        MOVAH,09
        MOV DX, OFFSET String
        Int 21h
        MOV AH,4CH
        INT 21H
MAIN ENDP
END MAIN
```

.