Day: M T W T F S Date: __/__/__

ASSIGNMENT #4

SUBJECT # ASSEMBLY LANGUAGE

I.D # 13137

SUBMITTED TO # SIR AMIN

(Q1) what will be the value of the destination operand after each of the following instructions execute in sequence?

```
.data
val1      WORD 1000h
val2      WORD 2000h
arrayB    BYTE 10h,20h,30h,40h,50h
arrayW    WORD 100h,200h,300h
arrayD    DWORD 10000h,20000h,
.code
mov  bx, 0A69Bh
movzx cx,bl          ;(a) Cx = 009Bh
movsx cx,bl          ;(b) Cx = 007Bh
mov   ax,val1
xchg  val2,ax        ;(c) val2 = 2000h, 1000h
```


Scanned with CamScanner

Day: M T W T F S Date: __/__/__

```
mov al, [arrayB+1]   ;(d) AL = 20h
mov ax, [arrayW+2]   ;(e) Ax = 2000h
mov eax, [arrayD+4]  ;(f) EAx = 20000h
```

Q2) write down the value of destination operands & flags after the execution of each instruction:

```
.code
mov cx, 1        Zero flag
Sub cx, 1        ;(a) Cx = 0      ZF = 1
mov cx, 0        Sign flag
Sub cx, 1        ;(b) Cx = -1     SF = 1
mov al, 0FFh     Carry flag
add al, 1        ;(c) AL = 00     CF = 1 Too big
mov al, 0
Sub al, 1        ;(d) AL = FF     CF = 1 below zero
mov al, 7Fh      Overflow flag
add al, 1        ;(e) AL = 80h    OF = 01
mov al, -128
neg al,          ;(f) CF = 1      OF = 1
```

Scanned with CamScanner

Day: M T W T F S

Date: __/__/__

(Q3) what will be the value of EAx after each of the following instruction execute?

```
.data
myBytes   BYTE 10h,20h,30h,40h
myWords   WORD 3 DUP(?),2000h
myString  BYTE "ABCDE"
.Code
mov eax, TYPE myBytes ; (a) EAx = 1
mov eax, LENGTHOF myBytes ; (b) EAx = 4
mov eax, SIZEOF myBytes ; (c) EAx = 4
mov eax, TYPE myWords ; (d) EAx = 2
mov eax, LENGTHOF myWords ; (e) EAx = 4
mov eax, SIZEOF myWords ; (f) EAx = 8
mov eax, SIZEOF myString ; (g) EAx = 5
```

(Q4) write down the value of each destination operands:

```
.data
val32 LABEL DWORD
varB BYTE 78h,56h,34h,12h
val8 LABEL BYTE
varD DWORD 12345678h
.code
mov bl, BYTE PTR varD ; (a) BL = 78h
mov eax, DWORD PTR varB ; (b) EAx = 78563412
mov al , val8 ; (c) AL =
mov eax , val32 ; (d) EAx =
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

Q5) what will be the value of the destination operand after each of the following instructions execute in sequence?

```
.data
myBytes BYTE 10h,20h,30h,40h
myWords WORD 8Ah,3Bh,72h,44h,66h
myDoubles DWORD 1,2,3,4,5
.code
mov esi, OFFSET myBytes
mov al,[esi+3]          ;(a) AL = 40h
mov esi OFFSET mywords+2
mov ax,[esi]            ;(b) Ax = 003Bh
mov edi, 8
mov edx, myDoubles[edi] ;(c) EDx = 3
```

Q6) write assembly language code for each of the following.

a) Convert the character in AL to upper case.

SOLUTION:-

```
.MODEL SMALL
.STACK 100H

.DATA
    PROMPT_1  DB 'Enter the lower case letter: $'
    PROMPT_2  DB 'The upper case letter is: $'

.Code
MAIN PROC
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

```
MOV AX, @DATA          ;initialize DS
MOV DS, AX

LEA DX, PROMPT_1       ;load & print PROMPT_1
MOV AH, 9
INT 21H

MOV AH, 1              ;read a letter
INT 21H

MOV BL, AL             ;Save the letter in BL

MOV AH, 2              ;return carriage
MOV DL, 0DH
INT 21H

MOV DL, 0AH            ;line feed
INT 21H

LEA DX, PROMPT_2       ;load & print PROMPT_2
MOV AH, 9
INT 21H

SUB BL, 20H            ;Convert a lower case
                       letter to upper case letter

MOV AH, 2              ;print the upper case letter
MOV DL, BL
INT 21H
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

```
MOV AH, 4CH      ;return control to DOS
INT 21H
MAIN ENDP
END MAIN
```

b) Convert a binary decimal byte into its equivalent ASCII decimal digit.

SOLUTION:-

Use the OR instruction to set bits 4 & 5.

```
mov al,6              ; AL = 00000110b
or al,00110000b       ; AL = 00110110b
```

The ASCII digit '6' = 00110110b

c) Jump to label L1 if bits 0, 1, & 3 in al are all set.

SOLUTION:-

Clear all bits except bit 0, 1 & 3. Then compare the result with 00001011 binary.

```
and al, 00001011b ; clear unwanted bits
cmp al, 00001011b ; check remaining bits
je L1             ; all set? jump to L1
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

Q7) write each of the following pseudocode in assembly language & explain:

a) if (var1 <= var2)
    var = 12B;
else
{
    var3 = 110;
    var4 = 90;
}

SOLUTION:-

```
mov ecx , var1
Cmp ecx , var2
jle L1
mov var3 , 110
mov var4 , 90
jmp L2
L1: mov var4 , 90
L2:
```

b) if (val1 > ecx) or (ecx > edx) then
X = 30
else
X = 40;

SOLUTION:-

```
Cmp val1, ecx
ja L1
mov eax, 40
Cmp ecx, edx
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

```
ja L1
mov x,40
jmp next
L1: mov x,30
next:
```

c) while (eax < ebx)
    eax = eax + 1;

SOLUTION:-

```
Cmp eax,ebx     ;check loop condition
jae next        ;false? exit loop
inc eax         ;body of loop
jmp top         ;repeat the loop
next:
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

Q8a) write a sequence of statements that use only PUSH & POP instructions to exchange the values in the EAx & EBx registers?

Ans) A sequence of statement are push ebx;
Assume Ebx = x & EAx = y, here the content of EBx (i.e x) is pushed. Push eax, which is assumed to be y pop ebx; y from stack is assigned to Ebx, therefore EBx = y.
Pop eax; x from stack is assigned to EAx therefore EAx = x.

b) write a program with a loop & indirect addressing that copies a string from source to target, reversing the character order in the process. Use the following variable:
Source BYTE "This is the source string", 0
target BYTE SIZEOF Source DUP ('#')

Ans)

SOLUTION:-

```
.386
.model flat .stdcall
.stack 4096
Exit Process PROTO , dw Exit code : DWORD

.data
Source BYTE "This is the source string", 0
target BYTE SIZEOF Source DUP ('#')
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

```
.code
main PROC
mov esi,0
mov edi, LENGTHOF source-1
mov ecx, SIZEOF source.

L1:
  mov eax, 0
  mov al, source [esi]
  mov target [edi], al
  inc esi
  dec edi
  loop L1

INVOKE ExitProcess, 0
main ENDP
END main
```

c) write a program that displays a string in all possible combinations of foreground & 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.

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

SOLUTION:-

```
INCLUDE Irvine32.inc

.data
Count DWORD?

.code
main PROC

    mov eax, 0 + (0*16)
    mov ecx, 16
    L1:
        mov Count, ecx
        push eax
        mov ecx, 16
    L2:
        Call SetTextColor
        push eax
        mov al, 'H'
        Call writechar
        pop eax
        inc eax
    Loop L2
    Call crlf
    pop eax
    add eax, 16
    mov ecx, Count
Loop L1
```

Scanned with CamScanner

Day: M T W T F S Date: __/__/__

```
    Call exit
    Call wastMsg
    exit
main ENDP


END main
```

Scanned with CamScanner