Department of Computer Engineering

Digital Hardware Systems

CpE 3104 - Microprocessors

Laboratory Exercise No.:	3	**Date Performed:**	
Laboratory Exercise Title:	Assembly Language Instructions		
Name of Student(s):	Christian Jay Y. Gallardo Jhon Fil Tizon	**Document Version:**	1

Laboratory Report

Activity #33-1

JMP HERE	AX=0000H BX=0000H CX=0031H DX=0000H
	IP=010EH

MOV DX, OFFSET BUFFER	AX=0000H BX=0000H CX=0031H DX=0102H
	IP=0111H

MOV AH, 0AH	AX=0A00H BX=0000H CX=0031H DX=0102H
	IP=0113H

INT 21H	AX=0A00H BX=0000H CX=0031H DX=0102H
	CS=F400H IP=0200H SP=FFF8H

CALL PRINT	AX=0A0DH BX=0000H CX=0031H DX=0102H
	CS=0700H IP=0115H SP=FFFEH

MOV AH, 04CH	AX=0924 BX=0001 CX=0031 DX=0104
	IP=0118 SP=FFFE

INT 021H	AX=4C24 BX=0001 CX=0031 DX=0104
	IP=011A

INT 020H	AX = 0924 BX=0001 CX=0031 DX=0104
	CS=F400 IP 0204

XOR BX, BX	AX=0A0DH BX=0000H CX=0031H DX=0102H
	CS=0700H IP=0120H SP=FFFCH

MOV BL, BUFFER[1]	AX=0A0DH BX=0005H CX=0031H DX=0102H
	CS=0700H IP=0124H SP=FFFCH

MOV BUFFER[BX+2], '$'	AX=0A0DH BX=0005H CX=0031H DX=0102H
	CS=0700H IP=0129H SP=FFFCH

MOV DX, OFFSET BUFFER + 2	AX=0A0DH BX=0005H CX=0031H DX=0104H
	CS=0700H IP=012CH SP=FFFCH

MOV AH, 9H	AX=090DH BX=0005H CX=0031H DX=0104H

INT 21H	AX=090D BX=0001 CX=0031 DX=0104 IP=012E
FINAL OUTPUT =	**CJAYCJAY**

Activity #33-2

CODE:

```
org 100h
jmp start

; ---------------------------
; DATA
; ---------------------------
msgName   db 13,10,'Enter complete name (end with dollarsign: $): $'
msgProg   db 13,10,'Enter program (end with dollar sign: $): $'
msgYear   db 13,10,'Enter year level (end with dollar sign: $): $'
msgTimes  db 13,10,'Enter number of times to display (0-9): $'
nl        db 13,10,'$'

nameBuf   db 80 dup(?)
progBuf   db 80 dup(?)
yearBuf   db 80 dup(?)
times     db 0
count     db 0          ; loop counter kept in memory (not in CX)

; ---------------------------
; CODE
; ---------------------------
start:
   ; DS = CS ; ES = DS
   push cs
   pop  ds
   mov  ax, ds
   mov  es, ax

   ; ---- NAME ----
   lea dx, msgName
   mov ah, 9
   int 21h
   lea di, nameBuf
   call read_until_dollar

   ; ---- PROGRAM ----
   lea dx, msgProg
   mov ah, 9
   int 21h
   lea di, progBuf
   call read_until_dollar
```

```
    ; ---- YEAR ----
    lea dx, msgYear
    mov ah, 9
    int 21h
    lea di, yearBuf
    call read_until_dollar

    ; ---- TIMES ----
    lea dx, msgTimes
    mov ah, 9
    int 21h

get_digit:
    mov ah, 01h
    int 21h             ; AL = key
    cmp al, 0Dh            ; ignore Enter
    je  get_digit
    cmp al, '0'
    jb  get_digit
    cmp al, '9'
    ja  get_digit
    sub al, '0'
    mov times, al
    mov count, al          ; copy to working counter

    ; ---- Display Loop using memory counter ----
display_loop:
    cmp byte ptr count, 0
    je  done

    ; print name
    lea dx, nameBuf
    mov ah, 9
    int 21h
    lea dx, nl
    mov ah, 9
    int 21h

    ; print program
    lea dx, progBuf
    mov ah, 9
    int 21h
    lea dx, nl
    mov ah, 9
    int 21h

    ; print year
    lea dx, yearBuf
    mov ah, 9
    int 21h
    lea dx, nl
    mov ah, 9
    int 21h

    ; delay between repeats
    call delay

    dec byte ptr count
    jmp display_loop

done:
    mov ah, 4Ch
    int 21h

; ---------------------------
; SUBROUTINES
; ---------------------------
; Read characters until '$' (stores bytes and appends '$')
read_until_dollar proc
next_char:
    mov ah, 01h
    int 21h              ; AL = key (echoed)
    cmp al, '$'
    je  end_input
    cmp al, 0Dh              ; ignore CR
    je  next_char
    cmp al, 0Ah              ; ignore LF
    je  next_char
    stosb                ; ES:DI <- AL
    jmp next_char
end_input:
    mov al, '$'
    stosb                ; terminator for AH=9
    ret
read_until_dollar endp

; simple software delay
delay proc
    mov bx, 0FFFFh
d1:
    nop
    dec bx
    jnz d1
    ret
delay endp
```