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 #36-1

Instruction	Register values after execution
MOV AX, OFFSET STRING	AX = 0102h
MOV SI, AX	SI = 0102h
CALL REVERSE	IP jumps to REVERSE, SP decremented, return address pushed
MOV CX, 0H	CX = 0000h
MOV AL, [SI]	AL = first char of STRING SI = 0103h
CMP AL, '$'	Flags updated
PUSH [SI]	Stack
INC SI	SI = 0103h
INC CX	CX = 0001h
JMP LOOP1	Loop back
MOV SI, OFFSET STRING	SI = 0102h
POP DX	DX = last pushed char
XOR DH, DH	DX = 0047h
MOV [SI], DL	First position of STRING replaced with 'G'
INC SI	SI = 0119
DEC CX	CX = 0000h
JMP LOOP2	Loop back until CX=0
MOV [SI], '$'	Final terminator placed
RET	Return to caller

Activity #36-2

; 36-2 Fixed & Optimized

```
org 100h

.data
   msgInput    db 13,10,'Enter a string: $'
   msgNoNum    db 13,10,'There is no number in the inputted string.$'
   msgWithNum  db 13,10,'The inputted string contains a number: $'
   msgPal      db 13,10,'The inputted string is a palindrome.$'
   msgNotPal   db 13,10,'The inputted string is NOT a palindrome.$'

   ; DOS 0Ah buffer: [0]=max,[1]=len,[2..]=data, then CR at [2+len]
   buffer      db 50,0, 50 dup(0)

   numFound    db 0
   numChar     db 0

.code
main proc
   mov ax, @data
   mov ds, ax

   ; --- prompt & read line ---
   mov ah, 09h
   lea dx, msgInput
   int 21h

   mov ah, 0Ah
   lea dx, buffer
   int 21h

   ; len = buffer[1], str = buffer+2, CR at [2+len]
   mov cl, [buffer+1]          ; CL = length (0..50)
   mov si, offset buffer+2     ; SI = start of string
   mov bx, si
   add bx, cx                  ; BX = &str[len]
   mov byte ptr [bx], '$'      ; replace CR with '$' for printing if needed
```

```
    ; --- scan for first digit '0'..'9' ---
    mov byte ptr numFound, 0
    mov di, si            ; DI = scan ptr
    mov ch, 0              ; CX=length (already CL=length); CH=0 ok

ScanDigit:
    cmp cl, 0
    je  DoneScanDigit
    mov al, [di]
    cmp al, '0'
    jb  NotDigit
    cmp al, '9'
    ja  NotDigit
    ; found a digit
    mov numFound, 1
    mov numChar, al
    jmp DoneScanDigit
NotDigit:
    inc di
    dec cl
    jmp ScanDigit

DoneScanDigit:
    ; print digit result
    cmp numFound, 1
    jne NoNumber
    mov ah, 09h
    lea dx, msgWithNum
    int 21h
    mov dl, numChar
    mov ah, 02h
    int 21h
    jmp DoPalindrome

NoNumber:
    mov ah, 09h
    lea dx, msgNoNum
    int 21h

; --- palindrome check (case-sensitive, includes all chars typed) ---
DoPalindrome:
    mov cl, [buffer+1]         ; CL = length
    mov si, offset buffer+2    ; SI = left
    mov ch, 0
    mov bx, si
    add bx, cx
    dec bx                     ; BX = &str[len-1]
    mov di, bx                 ; DI = right

PalLoop:
    ; stop when left >= right => palindrome
    mov ax, si
    cmp ax, di
    jae PalYes

    mov al, [si]
    mov bl, [di]
    cmp al, bl
    jne PalNo

    inc si
    dec di
    jmp PalLoop

PalYes:
    mov ah, 09h
    lea dx, msgPal
    int 21h
    jmp Done

PalNo:
    mov ah, 09h
    lea dx, msgNotPal
    int 21h

Done:
    mov ah, 4Ch
    int 21h
main endp
end main
```