3. Develop an assembly language program to reverse a given string and verify whether it is a *Palindrome* or not. Display the appropriate message

```
.model small

initds macro
    mov ax,@data        ; initializing the data segment
    mov ds,ax           ; it is ds, not dx
endm

inites macro
    mov es,ax           ; initializing the extra segment
endm

printf macro msg
    lea dx,msg          ; load the effective address to dx
    mov ah,9            ; function number is 9
    int 21h             ; using dos interrupt 21h
endm

getchar macro
    mov ah,1            ; this macro takes 1 key input,
    int 21h             ; its ascii value in hex stores in al
endm

exit macro
    mov ah,4ch          ; to terminate
    int 21h
endm

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

.data
      original db 30 dup(?)  ; 1st array
      reverse db 30 dup(?)   ; 2nd array to store the reversed array

      ask db 10,13,"String please:$"
      palindromemsg db 10,13,"Palindrome$"
      notpalindromemsg db 10,13,"Not Palindrome$"

.code

      initds

      inites       ; initializing extra segment (why??? b'coz we are
                   ; playing with strings)

      lea si, original ; 1st array starting index to si
      lea di, reverse  ; 2nd array starting index to di
      printf ask
      mov cx,00    ;counter..right now it's 0 (we haven't taken any i/p)

      takeinput:

            getchar      ; takes single character (pressed key's
                         ; ascii value goes to AL automatically)
            cmp al,13    ; compare with ENTER key
```

```
        je done       ; if you press ENTER key, then goto done
        mov [si],al   ; else, store your key in array
        inc cx        ; keeps the no. of elements in array
        inc si         ; move to next position
        jmp takeinput ; repeat till you press ENTER key

done: dec si      ; point to the last position

reversingtask:

        mov al,[si] ; last element of si
        mov [di],al ; put that to first element of di
        inc di      ; inc 2nd array position
        dec si      ; dec 1st array position
        jnz reversingtask

lea si, original ; comparison part
lea di, reverse
cld              ; clear direction flag
                 ; (so that si & di are auto incremented)
repe cmpsb       ; comparing [si] & [di]
je palin         ; if all the characters are equal, then goto palin

printf notpalindromemsg ; else, not palindrome case
exit                    ; bye bye!


palin: printf palindromemsg    ; palindrome
exit                           ; bye bye!

end
```