3.↳ WAP to check whether a string is palindrome or no

steps:-

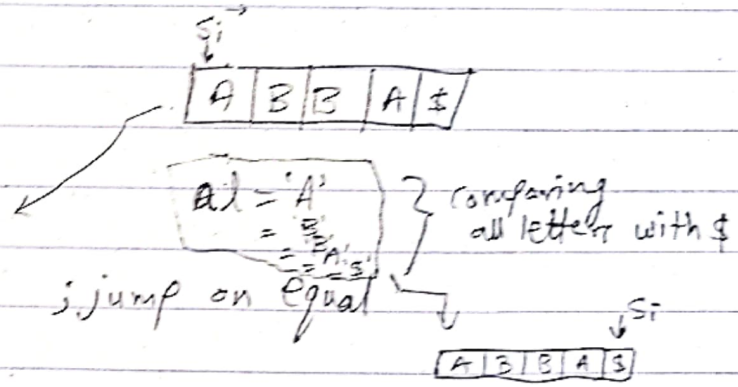
1. Create a string
2. Traverse to the end of string.
3. Get address of end of string in Si
4. Load starting add. of string in di
5. Compare value stored at address
6. Increment the pointer, Di
7. Decrements the pointer, Si
8. Compare again value stored at Si & di
9. Repeat the steps until Di <= Si.
10. If all characters match print string is palindrome.

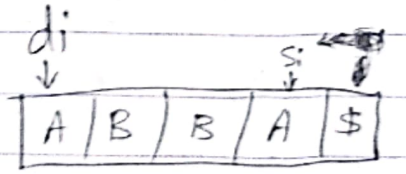
A	B	B	A	$

```
DATA SEGMENT
    String  db  'ABBA','$'
    String1 db  'String is palindrome','$'
    string2 db  'string is not palindrome','$'
ENDS

CODE SEGMENT
    START:
        MAIN PROC FAR
        mov ax, @data
        mov ds,ax
        CALL PALINDROME
        MOV ah, 4ch
```

(2)

```
int 21h
MAIN ENDP
CODE ENDS
END START

PALINDROME PROC
    mov si, offset string
Loop1:
    mov al,[si]
    cmp al, '$'
    JE  Label1
    inc si
    JMP Loop1
Label1:
    mov di, offset string
    dec si
Loop2:
    cmp si, di
    JL  output1
    mov al,[si]
    mov bl,[di]
    cmp al,bl
    JNE output2
    dec si
    inc di
    JMP Loop2
```





compare si & di until si is less than di.

al = A
bl = A

AL = B
Bl = B
:

until si < di

```
msg1 db 0dh, 0ah, 'character is vowel: $'
     ↑ carriage return   ↑ line return
```

```
output1:
        LEA dx, string1
        mov ah, 09h
        int 21h
        ret
output2:
        LEA dx, string2
        mov ah, 09h
        int 21h
        ret
PALINDROME ENDP
    END MAIN
```

4. Write an ALP in 8086 to check whether the entered ~~number~~ character is vowel or consonant.

```
.model small
.stack 100h
.data
    msg  db 'Enter a character: $'
    msg1 db 0dh, 0ah, 'character is Vowel $'
    msg2 db 0dh, 0ah, 'character is Consonant $'
.code
    main proc
        mov ax, @data
        mov ds, ax
    ; Display a message for input
    mov ah, 09h
    lea dx, msg
    int 21h
```

(4)

```
; input a character
    mov ah, 01h
    int 21h          ; al register contains input

  cmp al, 'a'
  je vowel
  cmp al, 'e'
  je vowel
  cmp al, 'i'
  je vowel
  cmp al, 'o'
  je vowel
  cmp al, 'u'
  je vowel

  mov ah, 09h        ; display msg2
  lea dx, msg2
  int 21h
  jmp exit

vowel:
    mov ah, 09h      ; display msg1
    lea dx, msg1
    int 21h
exit:
    mov ah, 4ch      ; terminate program
    int 21h
main endp
end main
```

(1)

eg. REPNE SCASB → AL will be compared with data in ES:DI until CX=0 or ZF=1

5. Write an 8086 ALP to search given character in a string.

REPNZ / REPNZ → used to repeat given inst. until CX=0 or zero Flag ZF=1

SCASB → Compares bytes at ES:DI with AL & sets flags according to result.

```
DATA SEGMENT
    mystr db "Ascol Campus"
    char db 's'
    count dw 12                ; count for 12 letter & space.

    str1 db "character not found. $"
    str2 db "character found. $"
DATA ENDS

CODE SEGMENT
    assume cs: CODE , ds: DATA
    start:
        mov ax, data
        mov ds, ax
        mov es, ax
        mov cx, count
        mov al, char
        REPNZ SCASB
        JE FOUND

Not_found:
        LEA dx, str1
        mov ah, 09h
        int 21h
        jmp exit

FOUND:
        LEA dx, str2
        mov ah, 09h
        int 21h

EXIT:
        mov ah, 4ch
        int 21h
CODE ENDS
END start
```