VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Microprocessors and Interfacing

(CSE – 3002)

LAB EXPERIMENT– 4

Name: **Vibhu Kumar Singh**

Reg. No: **19BCE0215**

Teacher: **Mr. Konguvel E.**

1. Write and execute ALP to prompt the user to input a string and reverse it and display on the screen.

Page No.
Date :

19BCE0215

Q1) 1) ALP:

```
print macro m
. mov ah, 09h
  mov dx, offset m
  int 21h
endm

.model small
.data
    empty db 10,13," $"
    str1 db 25,?,25,dup('$')
    str2 db 25,?,25,dup('$')
    mstring db 10,13,"Enter the string: $"
    mreverse db 10,13,"Revcised string: $"
.code

start:
    mov ax,@data.
    mov ds,ax

    print mstring
    call accept_String

    mov si, offset str1
    mov di, offset str2

    mov al,[si]
    mov [di], al
    inc si
    inc di
```

Page No.
Date :

19BCE0215

```
mov al,[si]
mov [di],al
inc si
inc di

mov cl,str1+1
mov ch,00
add si,cx
dec si

move_more:
    mov al,[si]
    mov [di],al
    dec si
    inc di
    dec cl
    jnz move_more

    print mreverse
    print str2+2
    print empty

exit:
    mov ah,4ch
    int 21h

accept proc near
    mov ah,01
    int 21h
    ret
accept endp
```

Page No.
Date :

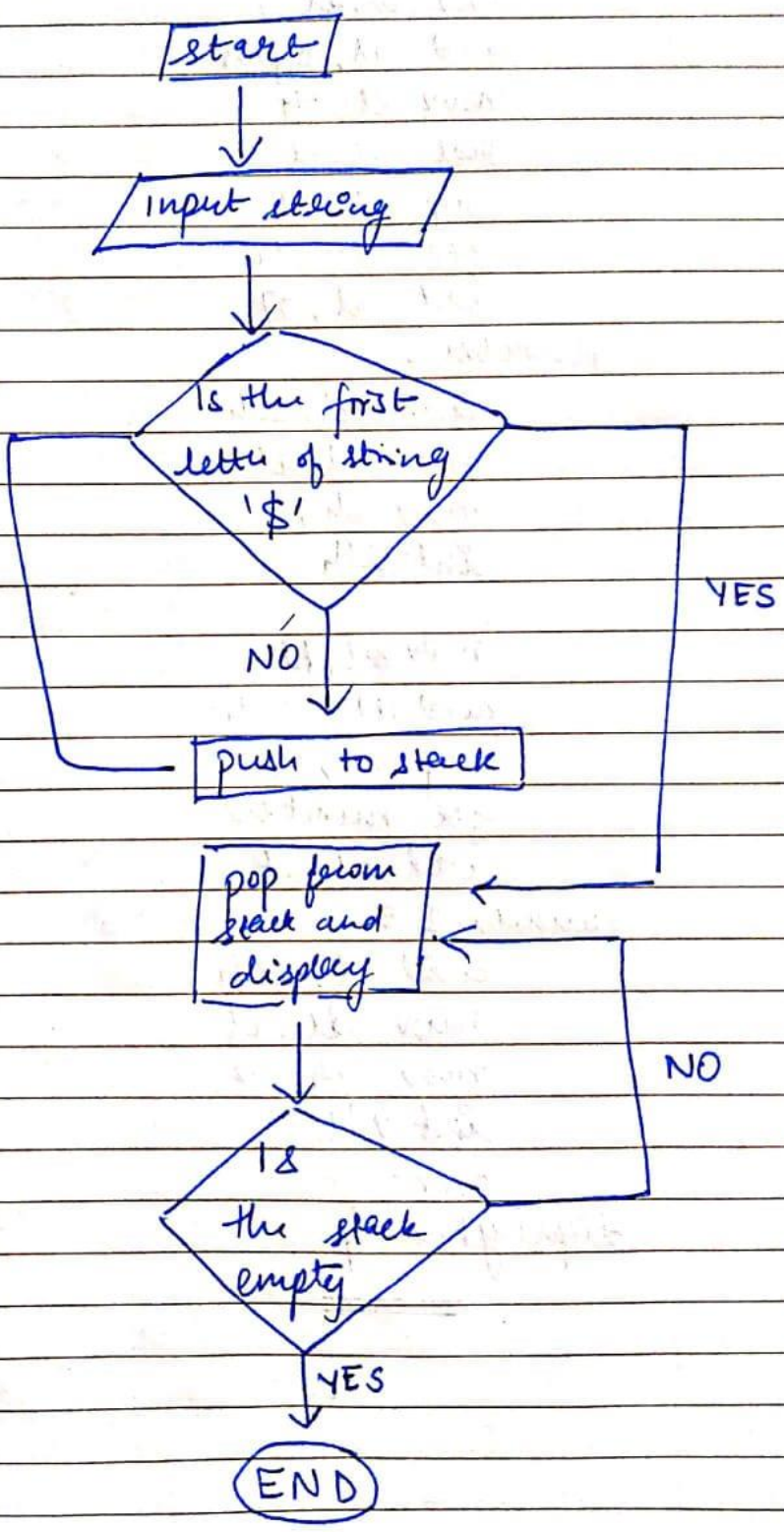
19BCE0215

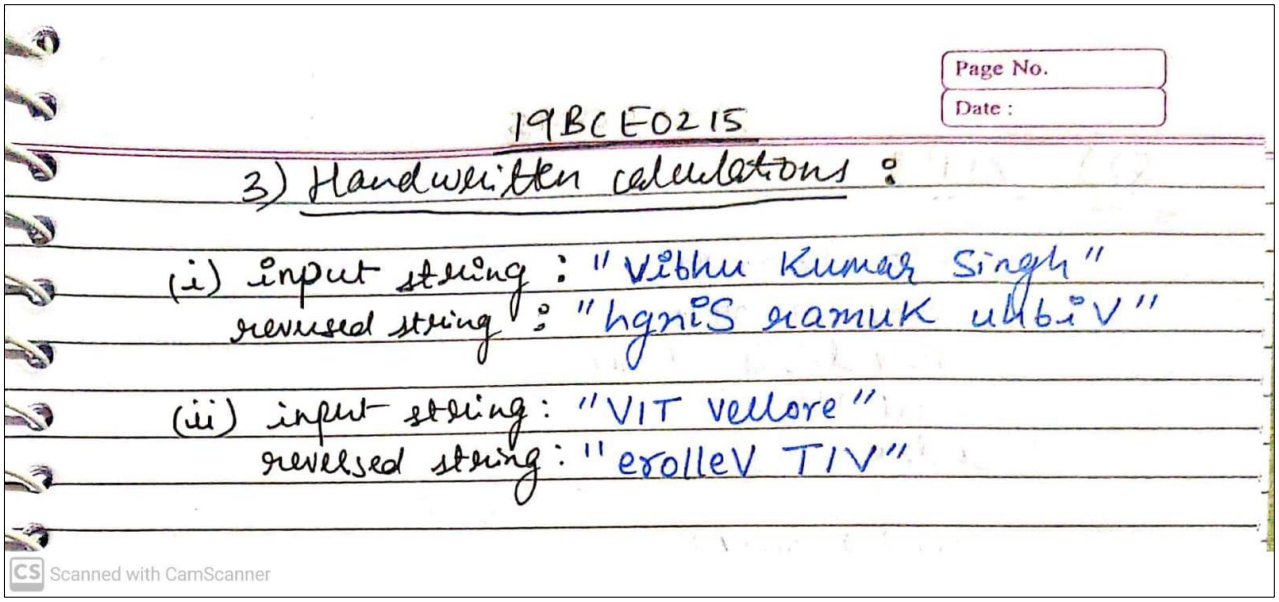
```
display1 proc near
        mov cl,bl
        mov bl,al
        and al, 0f0h
        mov cl, 04
        rol al,cl
        cmp al,09
        jbe number
        add al, 07
number:
        adel al, 30h
        mov dl,al
        mov ah,02
        int 21h

        mov al,bl
        and al,00fh
        cmp al,09
        jbe number2
        add al,07
number2:
        adel al, 30h
        mov dl,al
        mov ah,02
        int 21h
        ret
display1.endp.
```

Page No.
Date :

19BLE0215

2) Flowchart :





Screenshot of ALP:

edit: C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\Fall Semester 21-22\Micro\ELA\LAB-4\revstring.asm

file edit bookmarks assembler emulator math ascii codes help

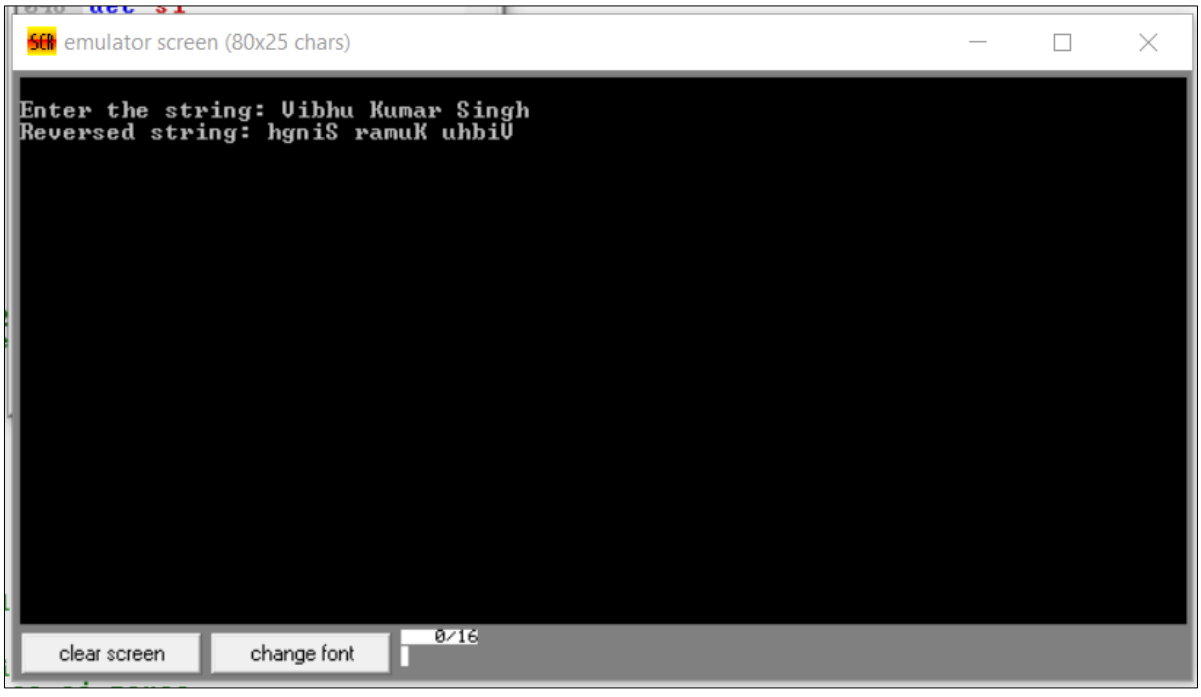
new open examples save compile emulate calculator convertor options help about

```
001 ;macro for printing a string
002 print macro m
003 mov ah,09h
004 mov dx,offset m
005 int 21h
006 endm
007
008 .model small
009 .data
010
011     empty db 10,13, "   $"
012     str1 db 25,?,25 dup('$')
013     str2 db 25,?,25 dup('$')
014     mstring db 10,13, "Enter the string: $"
015     mreverse db 10,13, "Reversed string: $"
016
017 .code
018
019 start:
020 mov ax,@data
021 mov ds,ax
022
023         print mstring
024         call accept_string
025
026
027         mov si,offset str1      ;point si to start of string1
028         mov di,offset str2      ;point di to start of string2
029
030
031
032         mov al,[si]          ;copy first two locations of string1 to string2
033         mov [di],al          ;since these contain the size and length of the string
034         inc si               ;which are same in reverse string also
035         inc di
036
037
038         mov al,[si]
039         mov [di],al
040         inc si
041         inc di
042
043         mov cl,str1+1 ;         copy length in cl
044         mov ch,00
045         add si,cx               ;add length of string1 to si to move it to last location
046         dec si                  ;si at last location of string1
047
048 move_more: mov al,[si]        ;copying character one by one from string1 pointed by si
049           mov [di],al         ; to string2 pointed by "di" in reverse order as si moves
050           dec si              ; from last character to first character
051           inc di
052           dec cl
053           jnz move_more
054
055
056         print mreverse
057         print str2+2      ;  printing the reversed string
058         print empty
```

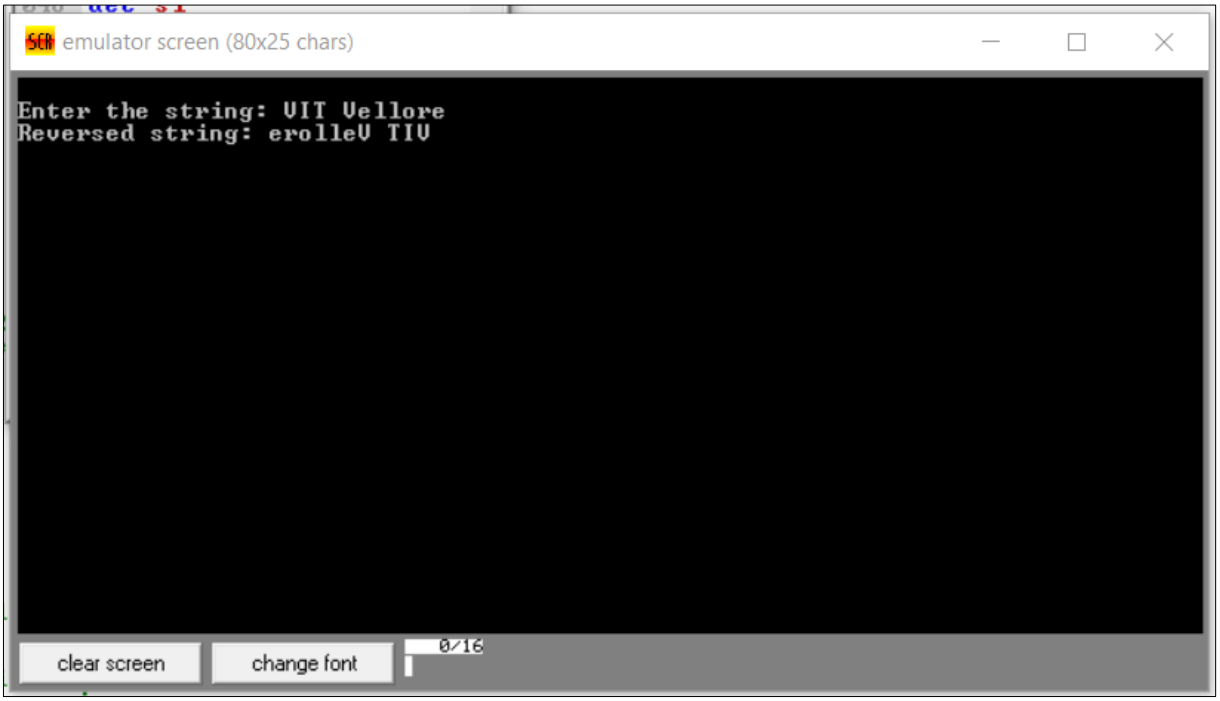
```
058          print empty
059
060  exit:
061  mov ah,4ch        ;exit the program
062  int 21h
063
064
065  ;accept procedure
066
067  accept proc near
068
069  mov ah,01
070  int 21h
071  ret
072  accept endp
073
074  display1 proc near
075
076     mov al,bl
077     mov bl,al
078     and al,0f0h
079     mov cl,04
080     rol al,cl
081
082     cmp al,09
083     jbe number
084     add al,07
085  number:   add al,30h
086            mov dl,al
087            mov ah,02
088            int 21h
089
090            mov al,bl
091            and al,00fh
092            cmp al,09
093            jbe number2
094            add al,07
095  number2:   add al,30h
096             mov dl,al
097             mov ah,02
098             int 21h
099  ret
100  display1 endp
101
102
103
104  accept_string proc near
105
106  mov ah,0ah           ;accept string from user function
107  mov dx,offset str1   ; store the string in memory pointed by "DX"
108  int 21h
109  ret
110  accept_string endp
111
112  end start
113  end
```

Screenshot of Output:

(i) Input: “Vibhu Kumar Singh”



(ii) Input: “VIT Vellore”



(Next Page)

2. Write and execute ALP to prompt the user to enter a string and check if it is a palindrome or not.

Page No.
Date :

19BCE0215

Q2) ALP :

```
print macro m
  mov ah, 09h
  mov dx, offset m
  int 21h
endm

.model small
.data
  empty db 10,13," $"
  str1 db 25,?,25 dup('$')
  str2 db 25,?,25 dup('$')
  len db ?
  mstring db 10,13,"Enter the string : $"
  notpalin db 10,13,"Not a palindrome $"
  palin db 10,13,"String is a palindrome $"
.code

start:
  mov ax,@data
  mov ds,ax

again:
  print mstring
  call accept_string

  mov si, offset string1
  mov cl, str1+1
  mov ch,00h
  mov len,cl
  inc si
```

Page No.
Date :

19BCE0215

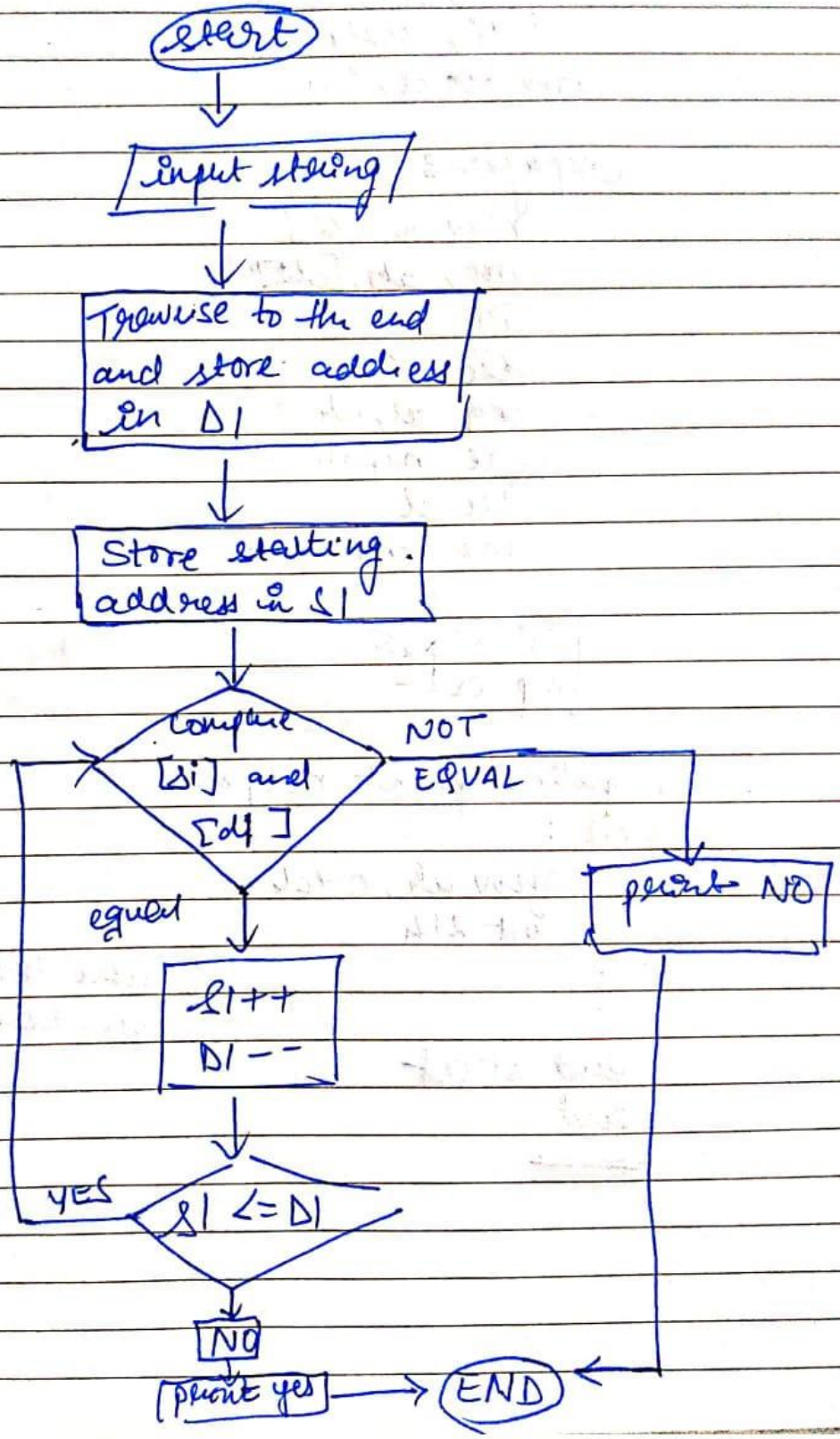
```
add si, cx

mov di, offset str1
add di, 002h
mov cl, len

cmpagain:
    mov al, [si]
    mov ah, [di]
    inc di
    dec si
    cmp al, ah
    jne nopalin
    dec cl
    jnz cmpagain

print palin
jmp exit

nopalin: print notpalin
exit:
    mov ah, 04ch
    int 21h
    :
    :
end start
end
```

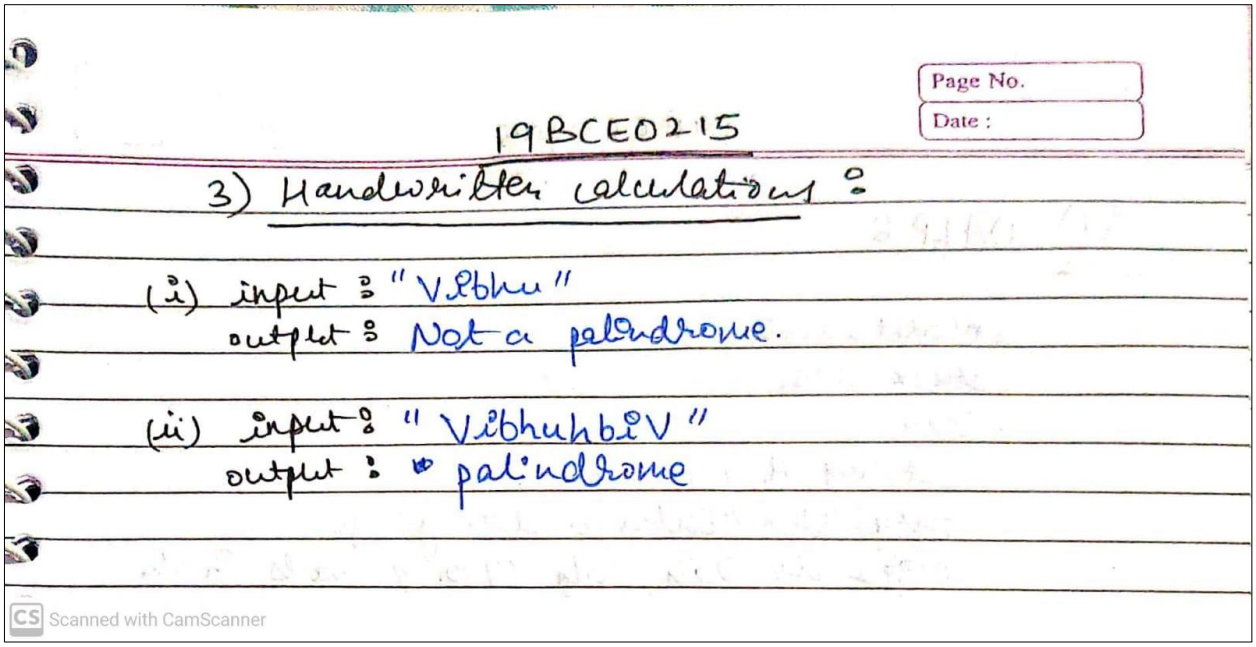
} user defined functions definitions.

Page No.
Date :

19BCE0215

2) Flowchart :





Screenshot of ALP:

edit: C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\Fall Semester 21-22\Micro\ELA\LAB-4\isPalindrome.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
;macro for printing a string
print macro m
mov ah,09h
mov dx,offset m
int 21h
endm

.model small


;******  Data Segment ******
.data

empty db 10,13, "   $"
str1 db 25,?,25 dup('$')
str2 db 25,?,25 dup('$')
len db ?
mstring db 10,13, "Enter the string: $"
notpalin db 10,13, "String is not a palindrome. $"
palin db 10,13, "String is a palindrome. $"


;********** Code Segment ************

.code

start:
mov ax,@data
mov ds,ax

again:

        print mstring
        call accept_string


        mov si,offset str1
        mov cl,str1+1          ;store the length
        mov ch,00h
        mov len,cl
        inc si
        add si,cx              ;si points to last

        mov di,offset str1     ;di to start of string
        add di,0002h           ;di to actual start of string;

        ;in string 0th byte->size
        ;          1st byte->length of string
        ;          from 2nd byte->actual string

        mov cl,len ;load counter
 cmpagain: mov al,[si]
           mov ah,[di]
           inc di
           dec si
           cmp al,ah
           jne nopalin
           dec cl
```

edit: C:\Users\Vibhu\OneDrive - vit.ac.in\Desktop\Fall Semester 21-22\Micro\ELA\LAB-4\isPalindrome.asm

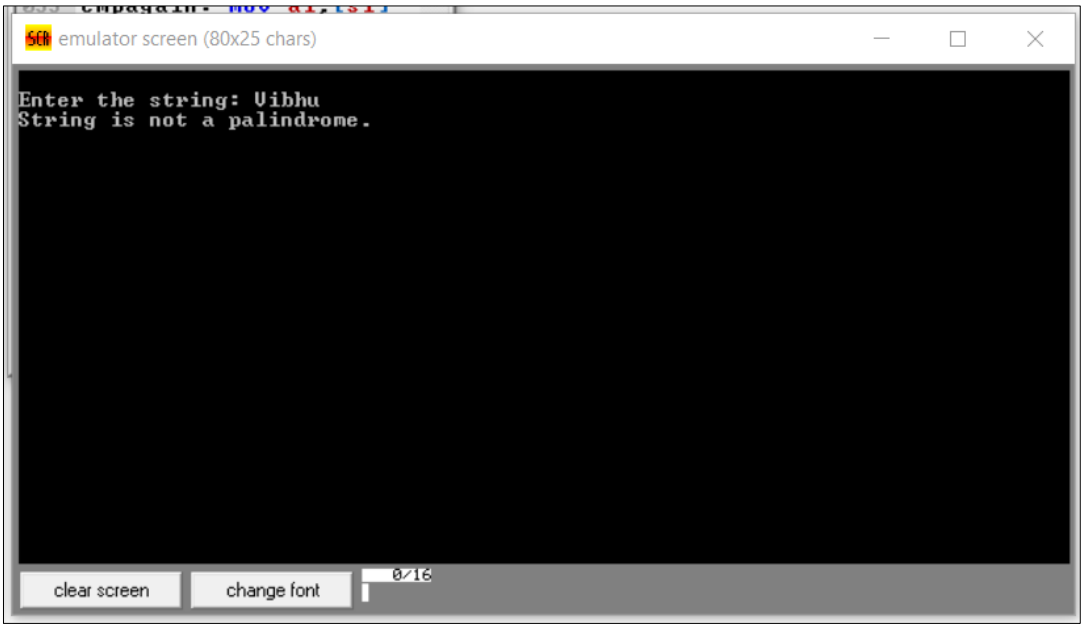
file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
061
062             print palin
063             jmp exit
064
065     nopalin: print notpalin
066
067     exit:
068     mov ah,4ch          ;exit the program
069     int 21h
070
071
072     ;accept procedure
073
074     accept proc near
075
076     mov ah,01
077     int 21h
078     ret
079     accept endp
080
081     display1 proc near
082
083        mov al,bl
084        mov bl,al
085        and al,0f0h
086        mov cl,04
087        rol al,cl
088
089        cmp al,09
090        jbe number
091        add al,07
092     number:  add al,30h
093              mov dl,al
094              mov ah,02
095              int 21h
096
097              mov al,bl
098              and al,00fh
099              cmp al,09
100              jbe number2
101              add al,07
102     number2:  add al,30h
103               mov dl,al
104               mov ah,02
105               int 21h
106     ret
107     display1 endp
108
109
110
111     accept_string proc near
112
113     mov ah,0ah           ;accept string from user function
114     mov dx,offset str1   ; store the string in memory pointed by "DX"
115     int 21h
116     ret
117     accept_string endp
118
119     end start
```

Screenshot of Output:

(i) Input: “Vibhu”



(ii) Input: “VibhuhbiV”