Lab 5

P1: Write a program to transfer the given string from source to destination using string instruction and also display the destination string.

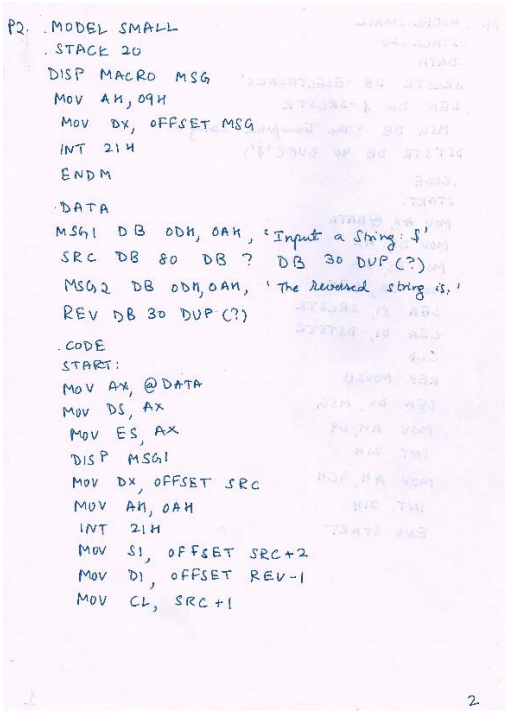
```
P1. .MODEL SMALL
    .STACK 20
    .DATA
    SRCSTR DB 'ELECTRONICS'
    LEN DW $-SRCSTR
     MSG DB 'The Transferred String='
    DSTSTR DB 40 DUP('$')
     .CODE
     START:
     MOV AX, @DATA
     MOV DS, AX
     MOV ES, AX
     MOV CX, LEN
      LEA SI, SRCSTR
      LEA DI, DSTSTR
      CLD
      REP MOVSB
      LEA DX, MSG
       MOV AH, 09
       INT 21H
      MOV AH, 4CH
       INT 21H
       END START
```

1

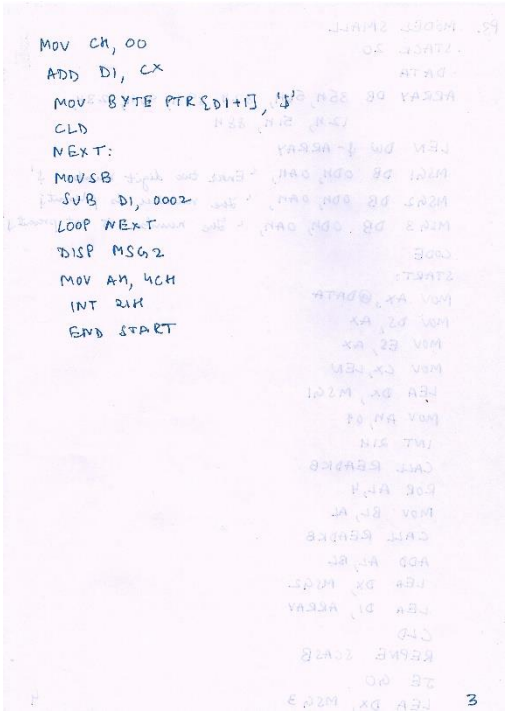
```
C:\TASM>lab5_1.exe
the transferred strings=electronics
C:\TASM>_
```

“Electronics” is transferred from SRCSTR to DSTSTR

P2: Write a program to read a string using DOS interrupts, reverse the entered string and display the same on the screen. Use MACRO for display.



```
P2. .MODEL SMALL
    .STACK 20
    DISP MACRO MSG
    MOV AH, 09H
     MOV DX, OFFSET MSG
     INT 21H
     ENDM
    .DATA
    MSG1 DB 0DH, 0AH, 'Input a String: $'
    SRC DB 80 DB ? DB 30 DUP(?)
    MSG2 DB 0DH, 0AH, 'The Reversed string is:'
    REV DB 30 DUP (?)
    .CODE
    START:
    MOV AX, @DATA
    MOV DS, AX
     MOV ES, AX
     DISP MSG1
     MOV DX, OFFSET SRC
     MOV AH, 0AH
      INT 21H
      MOV SI, OFFSET SRC+2
      MOV DI, OFFSET REV-1
      MOV CL, SRC+1
                                  2
```



```
MOV CH, 00
 ADD DI, CX
  MOV BYTE PTR[DI+1], '$'
  CLD
  NEXT:
  MOVSB
   SUB DI, 0002
   LOOP NEXT
   DISP MSG2
   MOV AH, 4CH
    INT 21H
    END START
                                  3
```

```
C:\TASM>lab5_2.exe

input a string:microprocessor
The reversed string is:rossecorporcim
C:\TASM>_
```

Entered string: microprocessor
Reversed string: rossecorporcim

P3: Write a program to read two digit decimal number using keyboard and search whether the number is present in an array or not. Display suitable message.

```
P3. MODEL SMALL
 .STACK 20
 .DATA
 ARRAY DB 35H, 56H, 82H, 89H, 90H, 23H,
          12H, 51H, 88H
 LEN DW $-ARRAY
 MSG1 DB ODH, OAH, 'Enter two digit number : $'
 MSG2 DB ODH, OAH, 'The number is present$'
 MSG3 DB ODH, OAH, 'The number is not present$'
 .CODE
 START:
 MOV AX, @DATA
 MOV DS, AX
  MOV ES, AX
  MOV CX, LEN
   LEA DX, MSG1
  MOV AH, 09
   INT 21H
   CALL READKB
   ROR AL, 4
   MOV BL, AL
    CALL READKB
    ADD AL, BL
     LEA DX, MSG2
     LEA DI, ARRAY
     CLD
     REPNE SCASB
     JE GO
     LEA DX, MSG3
```

4

```
C:\TASM>lab5_3.exe

Enter two digit numbers: 90
The number is present
```

90H is present, hence prints “The number is present”

```
GO:
 MOV AH, 09
 INT 21H
 MOV AH, 4CH
 INT 21H
 READKB PROC NEAR
 MOV AH, 01
  INT 21H
  CMP AL, 3AH
  JC SUB30
   SUB AL, 07H
   SUB 30:
    SUB AL, 30H
    RET
   READKB ENDP
   END START
```

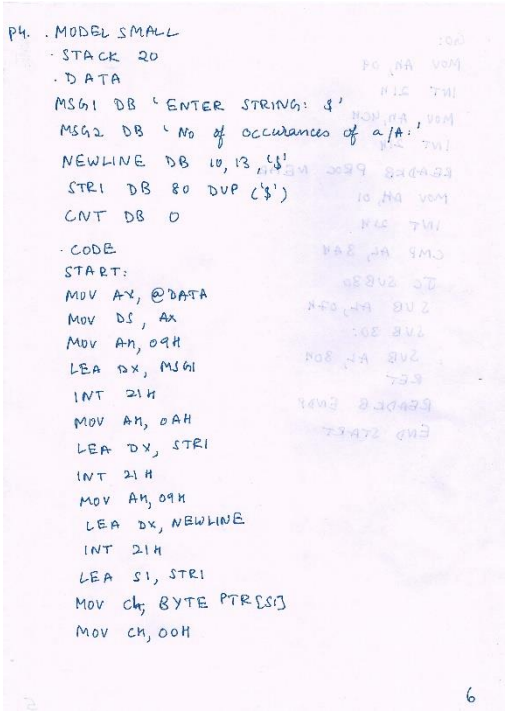
5

```
C:\TASM>lab5_3.exe

Enter two digit numbers: 18
The number is not present
```

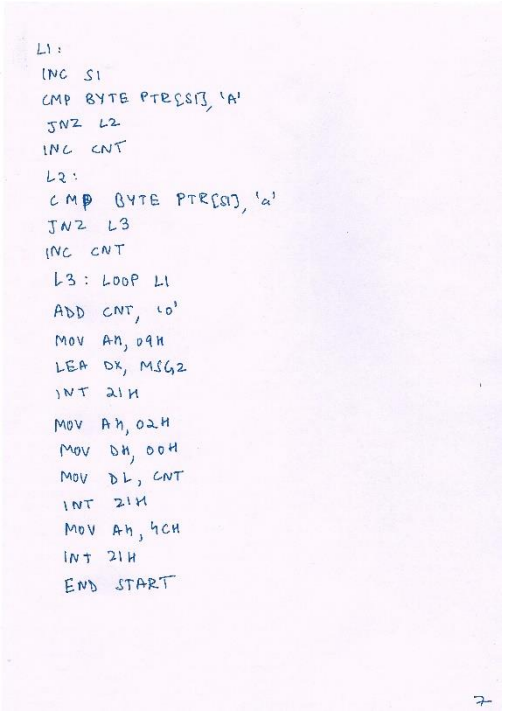
18H is present, hence prints “The number is not present”

P4: Write a program to count the number of occurrences of the letter “A” or “a” in the given message, and also count the number of words. Read the message using DOS interrupts.



```
P4. .MODEL SMALL
    .STACK 20
    .DATA
    MSG1 DB 'ENTER STRING: $'
    MSG2 DB 'No of occurances of a/A: '
    NEWLINE DB 10,13,'$'
    STR1 DB 80 DUP ('$')
    CNT DB 0
    .CODE
    START:
    MOV AX, @DATA
    MOV DS, AX
    MOV AH, 09H
    LEA DX, MSG1
    INT 21H
    MOV AH, 0AH
    LEA DX, STR1
    INT 21H
    MOV AH, 09H
    LEA DX, NEWLINE
    INT 21H
    LEA SI, STR1
    MOV CH, BYTE PTR[SI]
    MOV CH, 00H
```

6



```
L1:
INC SI
CMP BYTE PTR[SI], 'A'
JNZ L2
INC CNT
L2:
CMP BYTE PTR[SI], 'a'
JNZ L3
INC CNT
L3: LOOP L1
ADD CNT, '0'
MOV AH, 09H
LEA DX, MSG2
INT 21H
MOV AH, 02H
MOV DH, 00H
MOV DL, CNT
INT 21H
MOV AH, 4CH
INT 21H
END START
```

7

```
C:\TASM>lab5_ex.exe
Enter string :practice program
No of occurence of a or A :2
```