

# AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

## **Department of Computer Science and Engineering**

Course No : CSE 2214

Course Title : Assembly Language Programming Sessional

Assignment no : 11

**Date of Performance**: 09.09.2020

**Date of Submission**: 21.09.2020

Submitted To : Ms. Tahsin Aziz & Md. Siam Ansary

**Submitted By:** 

Group : A1

Name : Nusrat Jahan

**Id** : 18.01.04.020

**Section** : A

### Question no: 01

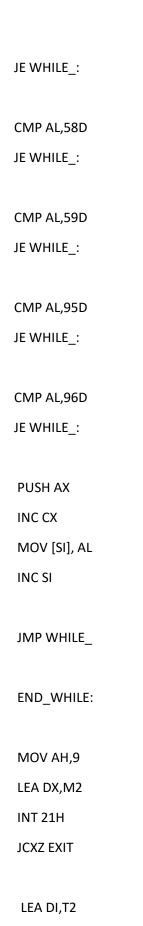
Question: Write a program that (a) lets the user input a string, (b) prints it forward and backward without punctuation and blanks on successive lines, and (c) decides whether it is palindrome and prints the conclusion.

# Solution:

```
.MODEL SMALL
.STACK 100H
.DATA
M1 DB OAH, ODH, 'Enter a string: ','$'
M2 DB 0AH,0DH,'Reversed string:','$'
M3 DB 0AH,0DH,'Forward string:','$'
M4 DB 0AH,0DH,'Backward string: ','$'
M5 DB 0AH,0DH,'The string is palindrome$'
M6 DB 0AH,0DH,'The string is not palindrome$'
T1 DB 100 DUP('$')
T2 DB 100 DUP('$')
.CODE
   MAIN PROC
     MOV AX,@DATA
     MOV DS,AX
     MOV ES,AX
     CLD
     MOV AH,9
       LEA DX,M1
       INT 21H
```

```
XOR CX,CX
      MOV AH,1
      LEA SI,T1
  WHILE_:
      INT 21H
      CMP AL,0DH
      JE END_WHILE
     CMP AL,33D
     JE WHILE_:
     CMP AL,34D
     JE WHILE_:
     CMP AL,39D
     JE WHILE_:
     CMP AL,''
     JE WHILE_:
     CMP AL,44D
      JE WHILE_:
      CMP AL,45D
JE WHILE_:
```

CMP AL,46D



# INC DI INT 21H LOOP TOP MOV AH,9 LEA DX,M3 INT 21H MOV AH,9 LEA DX,T1 INT 21H MOV AH,9 LEA DX,M4 INT 21H MOV AH,9 LEA DX,T2 INT 21H **CALL NEWLINE** LEA SI,T1

MOV BX,CX

MOV AH,2

TOP:

POP DX

MOV [DI],DL

JZ CHECK_PALINDROME
MOV AH,9
LEA DX,M6
INT 21H
JMP EXIT
CHECK_PALINDROME:
MOV AH,9
LEA DX,M5
INT 21H
EXIT:
MOV AH,4CH
INT 21H
MAIN ENDP
PROC NEWLINE
FROC NEWLINE

LEA DI,T2

MOV CX,BX

**PUSH AX** 

PUSH DX

REPE CMPSW

MOV AH,2

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H

POP DX

POP AX

RET

**NEWLINE ENDP** 

**END MAIN** 

### Question no: 02

Question: Write a program that reads a string STRING, a decimal integer S that represents a position in STRING, a decimal integer N that represents the number of bytes to be removed (both integers between 0 and 80), calls DELETE to remove N bytes at position S, and prints the resulting string.

# Solution:

```
.MODEL SMALL
.STACK 100H
.DATA
M1 DB OAH, ODH, 'Enter a string: ','$'
M2 DB 0AH,0DH,'The resulting string is:','$'
M3 DB OAH, ODH, 'Enter a decimal number S:','$'
M4 DB OAH, ODH, 'Enter a decimal number N:','$'
T1 DB 100 DUP('$')
.CODE
  MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        MOV ES,AX
        CLD
       MOV AH,9
       LEA DX,M1
```

```
MOV AH,1
  LEA SI,T1
  WHILE:
     INT 21H
     CMP AL,ODH
     JE END_WHILE
     MOV [SI], AL
     INC SI
     INC CX
 JMP WHILE
 END_WHILE:
    MOV AH,9
    LEA DX,M3
    INT 21H
CALL INDEC
   MOV BX,AX
   SUB BX,1
   MOV AH,9
   LEA DX,M4
```

INT 21H

XOR CX,CX

### INT 21H

# CALL INDEC CALL NEWLINE LEA DI,T1 ADD DI,BX SUB CX,BX SUB CX,AX LEA SI,T1 ADD SI,BX ADD SI,AX REP MOVSB

MOV AH,9

MOV [DI],'\$'

LEA DX,T1

INT 21H

MOV AH,4CH

INT 21H

MAIN ENDP

### PROC NEWLINE

PUSH AX	
PUSH DX	
MOV AH,2	
MOV DL,0DH	
INT 21H	
MOV DL,0AH	
INT 21H	
POP DX	
POP AX	
RET	
NEWLINE ENDP	
INDEC PROC	
PUSH BX	
PUSH CX	
PUSH DX	
BEGIN:	
BEGIN:	
BEGIN: XOR BX,BX	
XOR BX,BX	

INT 21H

REPEAT:

CMP AL,'0'

JNGE NOT\_DIGIT

CMP AL,'9'

JNLE NOT\_DIGIT

AND AX,000FH

**PUSH AX** 

MOV AX,10

MUL BX

POP BX

ADD BX,AX

MOV AH,1

INT 21H

CMP AL,0DH

JNE REPEAT

MOV AX,BX

EXIT:

POP DX

POP CX

POP BX

RET

NOT\_DIGIT:

MOV AH,2

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H

JMP BEGIN

RET

INDEC ENDP

END MAIN