



Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Course No : CSE2214

Course Title : Assembly Language Programming Sessional

Assignment No : 11

Date of Performance : 14-03-2021

Date of Submission : 21-03-2021

Submitted To : Ms. Tahsin Aziz & Mr. A.K.M. Amanat Ullah

Submitted By-

Group : B₂

Name : S. M Tasnimul Hasan

Id : 180204142

Section : B

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 a palindrome and prints the conclusion.

Answer:

.MODEL SMALL

.STACK 100H

.DATA

MSG1 DB 0AH,0DH,'Enter a string : ','\$'

MSG2 DB 0AH,0DH,'The reversed string is : ','\$'

MSG3 DB 0AH,0DH,'The forward string is : ','\$'

MSG4 DB 0AH,0DH,'The backward string is : ','\$'

PRINT_PALINDROME DB 0AH,0DH,'The string is palindrome\$'

PRINT_NOT_PALINDROME DB 0AH,0DH,'The string is not palindrome\$'

TEXT1 DB 100 DUP('\$')

TEXT2 DB 100 DUP('\$')

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV ES,AX

CLD

MOV AH,9

LEA DX,MSG1

INT 21H

XOR CX,CX

MOV AH,1

LEA SI,TEXT1

```
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  
JE WHILE_:
```

```
CMP AL,58D  
JE WHILE_:
```

```
CMP AL,59D  
JE WHILE_:
```

CMP AL,95D
JE WHILE_:

CMP AL,96D
JE WHILE_:

PUSH AX
INC CX
MOV [SI], AL
INC SI

JMP WHILE_

END_WHILE:

MOV AH,9
LEA DX,MSG2
INT 21H

JCXZ EXIT ; if CX register is 0

LEA DI,TEXT2
MOV BX,CX

MOV AH,2
TOP:
POP DX
MOV [DI],DL
INC DI
INT 21H
LOOP TOP

MOV AH,9
LEA DX,MSG3
INT 21H

```
MOV AH,9  
LEA DX,TEXT1  
INT 21H
```

```
MOV AH,9  
LEA DX,MSG4  
INT 21H
```

```
MOV AH,9  
LEA DX,TEXT2  
INT 21H
```

```
CALL NEWLINE
```

```
LEA SI,TEXT1  
LEA DI,TEXT2
```

```
MOV CX,BX  
REPE CMPSW
```

```
JZ PALINDROME
```

```
MOV AH,9  
LEA DX,PRINT_NOT_PALINDROME  
INT 21H
```

```
JMP EXIT  
PALINDROME:
```

```
MOV AH,9  
LEA DX,PRINT_PALINDROME  
INT 21H
```

```
EXIT:
```

```
MOV AH,4CH
```

INT 21H
MAIN ENDP

NEWLINE PROC

PUSH AX
PUSH DX

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.

Answer:

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 0AH,0DH,'Enter a string : ','$'
MSG2 DB 0AH,0DH,'The resulting string is : ','$'
MSG3 DB 0AH,0DH,'Enter a decimal number S : ','$'
MSG4 DB 0AH,0DH,'Enter a decimal number N : ','$'
```

```
TEXT1 DB 100 DUP('$')
```

```
.CODE
MAIN PROC
    MOV AX,@DATA
    MOV DS,AX
    MOV ES,AX
```

```
    CLD
```

```
    MOV AH,9
    LEA DX,MSG1
    INT 21H
```

```
    XOR CX,CX
```

```
    MOV AH,1
```

```
    LEA SI,TEXT1
```

```
    WHILE_:
```

```
INT 21H
CMP AL,0DH
JE END_WHILE
```

```
MOV [SI], AL
INC SI
INC CX
```

```
JMP WHILE_
```

```
END_WHILE:
```

```
MOV AH,9
LEA DX,MSG3
INT 21H
```

```
CALL INDEC
MOV BX,AX
SUB BX,1
```

```
MOV AH,9
LEA DX,MSG4
INT 21H
```

```
CALL INDEC
```

```
CALL NEWLINE
```

```
LEA DI,TEXT1
ADD DI,BX
```

```
SUB CX,BX
SUB CX,AX
```



```
LEA SI,TEXT1
ADD SI,BX
ADD SI,AX

REP MOVSB

MOV [DI],'$'

MOV AH,9
LEA DX,TEXT1
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:
```

```
XOR BX,BX  
XOR CX,CX  
MOV AH,1  
INT 21H
```

```
REPEAT2:  
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 REPEAT2
```

```
CMP AL,0  
JL REPEAT2  
CMP AL,80  
JG REPEAT2
```

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