



## **Ahsanullah University of Science & Technology**

### Department of Computer Science and Engineering

Course No : CSE 2214  
Course Title : Assembly Language Programming Sessional  
Assignment no : 11  
Date of Performance : 09-09-20  
Date of Submission : 22-09-20  
Submitted To : Ms.Tahsin Aziz & Md.Siam Ansary

#### Submitted By:

Name : Mahin opu  
ID : 17.02.04.006  
Year : 2<sup>nd</sup>  
Semester : 2<sup>nd</sup>  
Group : A1  
Section : A

**Question 01:**

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.

**Answer:****DATA SEGMENT**

**MSG1 DB 10,13,'ENTER ANY STRING :- \$'**

**MSG2 DB 10,13,'ENTERED STRING IS :- \$'**

**MSG3 DB 10,13,'LENGTH OF STRING IS :- \$'**

**MSG4 DB 10,13,'NO, GIVEN STRING IS NOT A PALINDROME \$'**

**MSG5 DB 10,13,'THE GIVEN STRING IS A PALINDROME \$'**

**MSG6 DB 10,13,'REVERSE OF ENTERED STRING IS :- \$'**

**P1 LABEL BYTE**

**M1 DB 0FFH**

**L1 DB ?**

**P11 DB 0FFH DUP ('\$')**

**P22 DB 0FFH DUP ('\$')**

**DATA ENDS****DISPLAY MACRO MSG**

**MOV AH,9**

**LEA DX,MSG**

**INT 21H**

**ENDM**

**CODE SEGMENT**

**ASSUME CS:CODE,DS:DATA**

**START:**

**MOV AX,DATA**

**MOV DS,AX**

**DISPLAY MSG1**

**LEA DX,P1**

**MOV AH,0AH**

**INT 21H**

**DISPLAY MSG2**

**DISPLAY P11**

**DISPLAY MSG3**

**MOV DL,L1**

**ADD DL,30H**

**MOV AH,2**

**INT 21H**

**DISPLAY MSG6**

**LEA SI,P11**

**LEA DI,P22**

**MOV DL,L1**

**DEC DL**

**MOV DH,0**

**ADD SI,DX**

**MOV CL,L1**

**MOV CH,0**

**REVERSE:**

**MOV AL,[SI]**

**MOV [DI],AL**

**INC DI**

**DEC SI**

**LOOP REVERSE**

**DISPLAY P22**

**LEA SI,P11**

**LEA DI,P22**

**MOV CL,L1**

**MOV CH,0**

**CHECK:**

**MOV AL,[SI]**

**CMP [DI],AL**

**JNE NOTPALIN**

**INC DI**

**INC SI**

**LOOP CHECK**

**DISPLAY MSG5**

**JMP EXIT**

**NOTPALIN:**

**DISPLAY MSG4**

**EXIT: MOV AH,4CH**

**INT 21H**

**CODE ENDS**

**END START**

**Question 02:**

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**

**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**