**AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY**

****

**Department of Computer Science and Engineering**

Program: BSc in Computer Science and Engineering

Course Code: CSE 2214

Assignment No: 04

Date of Submission: 18.08.2025

Submitted by,

Name: Danial Hossain Dani

Student ID: 20230104058

Lab Section: B1

**1. Write a program that lets the user type some text, consisting of words**

**separated by blanks, ending with a carriage return, and displays the**

**text in the same word order as entered, but with the letters in each**

**word reversed.**

**Answer:**

**.MODEL SMALL**

**.STACK 100H**

**.DATA**

**M1 DB 'Enter the string : $'**

**M2 DB 0DH,0AH,'The string with words reversed : $'**

**BUF DB 100 DUP('$')**

**LN DB 0DH,0AH,'$'**

**.CODE**

**P0 PROC**

**MOV AX, @DATA**

**MOV DS, AX**

**LEA DX, M1**

**MOV AH, 9**

**INT 21H**

**XOR CX, CX**

**LEA SI, BUF**

**L0:**

**MOV AH, 1**

**INT 21H**

**CMP AL, 0DH**

**JE L1**

**MOV [SI], AL**

**INC SI**

**INC CX**

**JMP L0**

**L1:**

**MOV BYTE PTR [SI], 0**

**LEA DX, M2**

**MOV AH, 9**

**INT 21H**

**LEA SI, BUF**

**L2:**

**MOV AL, [SI]**

**CMP AL, 0**

**JE L9**

**CMP AL, ' '**

**JNE L3**

**MOV DL, ' '**

**MOV AH, 2**

**INT 21H**

**INC SI**

**JMP L2**

**L3:**

**MOV DI, SI**

**L4:**

**MOV AL, [DI]**

**CMP AL, 0**

**JE L5**

**CMP AL, ' '**

**JE L5**

**INC DI**

**JMP L4**

**L5:**

**DEC DI**

**L6:**

**CMP DI, SI**

**JB L7**

**MOV DL, [DI]**

**MOV AH, 2**

**INT 21H**

**DEC DI**

**JMP L6**

**L7:**

**MOV AL, [SI]**

**CMP AL, 0**

**JE L9**

**L8:**

**MOV AL, [SI]**

**CMP AL, 0**

**JE L2**

**CMP AL, ' '**

**JE L2**

**INC SI**

**JMP L8**

**L9:**

**MOV AH, 4CH**

**INT 21H**

**P0 ENDP**

**END P0**

**2.** **Write a program that lets the user type in an algebraic expression,**

**ending with a carriage return, that contains round (parentheses),**

**square, and curly brackets. As the expression is being typed in, the**

**program evaluates each character. If at any point the expression is**

**incorrectly bracketed (too many right brackets or a mismatch between**

**left and right brackets), the program tells the user to start over. After**

**the carriage return is typed, if the expression is correct, the program**

**displays "expression is correct." If not, the program displays "too**

**many left brackets". In both cases, the program asks the user if he or**

**she wants to continue. If the user types 'Y', the program runs again.**

**Your program does not need to store the input string, only check it for**

**correctness.**

**Answer:**

**.MODEL SMALL**

**.STACK 100H**

**.DATA**

**P1 DB 0DH,0AH,'Enter an Algebraic Expression : $'**

**P2 DB 0DH,0AH,'Expression is Correct.$'**

**P3 DB 0DH,0AH,'Too many Left Brackets.$'**

**P4 DB 0DH,0AH,'Too many Right Brackets. Begin Again!$'**

**P5 DB 0DH,0AH,'Bracket Mismatch. Begin Again!$'**

**P6 DB 0DH,0AH,'Type Y if you want to Continue : $'**

**.CODE**

**X0 PROC**

**MOV AX,@DATA**

**MOV DS,AX**

**Q0:**

**LEA DX,P1**

**MOV AH,9**

**INT 21H**

**XOR CX,CX**

**Q1:**

**MOV AH,1**

**INT 21H**

**CMP AL,0DH**

**JE Q2**

**CMP AL,'('**

**JE Q3**

**CMP AL,'{'**

**JE Q3**

**CMP AL,'['**

**JE Q3**

**CMP AL,')'**

**JE Q4**

**CMP AL,'}'**

**JE Q5**

**CMP AL,']'**

**JE Q6**

**JMP Q1**

**Q3:**

**PUSH AX**

**INC CX**

**JMP Q1**

**Q4:**

**CMP CX,0**

**JLE Q8**

**POP DX**

**DEC CX**

**CMP DL,'('**

**JNE Q7**

**JMP Q1**

**Q5:**

**CMP CX,0**

**JLE Q8**

**POP DX**

**DEC CX**

**CMP DL,'{'**

**JNE Q7**

**JMP Q1**

**Q6:**

**CMP CX,0**

**JLE Q8**

**POP DX**

**DEC CX**

**CMP DL,'['**

**JNE Q7**

**JMP Q1**

**Q2:**

**CMP CX,0**

**JNE Q9**

**LEA DX,P2**

**MOV AH,9**

**INT 21H**

**JMP Q10**

**Q7:**

**LEA DX,P5**

**MOV AH,9**

**INT 21H**

**JMP Q10**

**Q9:**

**LEA DX,P3**

**MOV AH,9**

**INT 21H**

**JMP Q10**

**Q8:**

**LEA DX,P4**

**MOV AH,9**

**INT 21H**

**JMP Q10**

**Q10:**

**LEA DX,P6**

**MOV AH,9**

**INT 21H**

**MOV AH,1**

**INT 21H**

**CMP AL,'Y'**

**JE Q0**

**MOV AH,4CH**

**INT 21H**

**X0 ENDP**

**END X0**