Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Course No : CSE2214
Course Title : Assembly Language Programming Sessional
Assignment No : 08

Date of Performance : 19.01.22
Date of Submission : 02.02.22

Submitted To : Mr. Sajib Kumar Saha Joy & Zarin Tasnim Shejuti

Submitted By
Group : B1
Name : Kazi Atiqur Rahman
Id : 190204086
Section : B

Question No: 01

Question: 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
INPUT DB 'PLEASE ENTER SOME TEXT(SEPARATE BY
      BLANKS END WITH CARRIAGE RETURN)$'
REVERSE DB 0DH,0AH, 'REVERSER FORM : $'
COUNT DW 0

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

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

XOR CX, CX
MOV AH, 1

NO_ENTER:
INT 21H
CMP AL, 0DH
JE ENTER_PRESSED

PUSH AX
INC CX
JMP NO_ENTER

ENTER_PRESSED:
MOV BX, 50H
XCHG BX, SP
PUSH 0020H

XCHG BX, SP
INC COUNT

LOOP_1:
POP DX
XCHG BX, SP
```

```
PUSH DX
XCHG BX,SP
INC COUNT
LOOP LOOP_1

LEA DX, REVERSE
MOV AH,9
INT 21H

MOV CX,COUNT
MOV COUNT,0

PUSH 0020H
INC COUNT

RESULT:
XCHG BX,SP
POP DX.
XCHG BX,SP

CMP DL,20H
JNE NO_SPACE

MOV AH,2
```

```
LOOP_2:
POP DX
INT 21H

DEC COUNT
JNZ LOOP_2

MOV DX, 0020H

NO_SPACE:
PUSH DX
INC COUNT
LOOP RESULT

EXIT:
MOV AH, 4CH
INT 21H
MAIN ENDP

END MAIN
```

Question No: 02

Question: Write a Program that lets the user type in an algebraic expression, ending with a carriage return, that contains round, 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
INPUT_MSG DB 0DH,0AH,'ENTER AN ALGEBRIC EXPRE-
          SSION: $'
CORRECT_MSG DB 0DH,0AH,'EXPRESSION IS CORRECT.$'
MISMATCH_MSG DB 0DH,0AH,'MISMATCH BETWEEN LEFT
             AND RIGHT BRACKETS.$'
LEFT_MSG DB 0DH,0AH,'TOO MANY LEFT BRACKETS.$'
RIGHT_MSG DB 0DH,0AH,'TOO MANY RIGHT BRACKETS.$'
CONTINUE_MSG DB 0DH,0AH,'TYPE Y TO RUN AGAIN:$'

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

    START:
    LEA DX,INPUT_MSG
    MOV AH,9
    INT 21H
```

```
XOR CX, CX
MOV AH, 1

INPUT:
INT 21H

CMP AL, 0DH
JE END_INPUT

CMP AL, "["
JE PUSH_BRACKET

CMP AL, "{"
JE PUSH_BRACKET

CMP AL, "("
JE PUSH_BRACKET

CMP AL, "]"
JE SQUARE_BRACKET

CMP AL, "}"
JE CURLY_BRACKET

CMP AL, ")"
JE ROUND_BRACKET

JMP INPUT
```

```
PUSH_BRACKET:
PUSH AX
INC CX
JMP INPUT

ROUND_BRACKET:
POP BX
DEC CX

CMP CX,0
JL RIGHT_BRACKETS

CMP BL,"("
JNE MISMATCH
JMP INPUT

CURLY_BRACKET:
POP BX
DEC CX

CMP CX,0
JL RIGHT_BRACKETS

CMP BL,"{"
JNE MISMATCH
JMP INPUT
```

```
SQUARE_BRACKETS:
POP BX
DEC CX

CMP CX,0
JL RIGHT_BRACKETS

CMP BL,"["
JNE MISMATCH
JMP INPUT

END_INPUT:
CMP CX,0
JNE LEFT_BRACKETS

LEA DX, CORRECT_MSG
MOV AH,9
INT 21H

LEA DX, CONTINUE_MSG
INT 21H.

MOV AH,1
INT 21H

CMP AL,"Y"
JNE EXIT
JMP START
```

```
RIGHT_BRACKETS:
LEA DX, RIGHT_MSG
MOV AH,9
INT 21H
JMP START

MISMATCH:
LEA DX, MISMATCH_MSG
MOV AH,9
INT 21H
JMP START

LEFT_BRACKETS:
LEA DX, LEFT_MSG
MOV AH,9
INT 21H

LEA DX, CONTINUE_MSG
INT 21H

MOV AH,1
INT 21H

CMP AL, "Y"
JNE EXIT
JMP START

EXIT:
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```