**LAB-5**

**3. Replace each uppercase letter in the following string by its lowercase equivalent.**

**Use index addressing mode.**

**MSG DB “THIS IS CSE 331 LAB CLASS 5$”**

ORG 100H

.DATA

MSG DB "THIS IS CSE331 LAB CLASS 5$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV BX, 0

CHECK:

CMP MSG[BX],'$'

JE DISPLAY

CMP MSG[BX], 41H

JAE CHECK2

INC BX

JMP CHECK

CHECK2:

CMP MSG[BX], 5AH

JBE CONVERT

INC BX

JMP CHECK

CONVERT:

ADD MSG[BX], 20H

INC BX

JMP CHECK

DISPLAY:

MOV AH, 9

LEA DX, MSG

INT 21H

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**4. Write a program that will prompt the user to enter a hex digit character**

**(“0”.......”9” or “A”..........”F”), display it on the next line in decimal.**

**Sample 1:**

**Enter a Hex digit: B**

**The decimal value of B is 11**

**Sample 2:**

**Enter a Hex digit: 3**

**The decimal value of 3 is 3**

ORG 100H

.DATA

MSG1 DB "Enter a Hex digit: $"

MSG2 DB " The decimal value is: $"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

INPUT:

MOV AH, 9

LEA DX, MSG1

INT 21H

MOV AH, 1

INT 21H

MOV BL, AL

NEWLINE:

MOV AH, 2

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

CHECK\_ATOF:

CMP BL, 41H

JB CHECK\_1TO9

CMP BL, 46H

JA EXIT

JMP CONVERT

CHECK\_1TO9:

CMP BL, 30H

JB EXIT

CMP BL, 39H

JA EXIT

MOV AH, 9

LEA DX, MSG2

INT 21H

MOV AH, 2

MOV DL, BL

INT 21H

JMP EXIT

CONVERT:

SUB BL, 11H

MOV AH, 9

LEA DX, MSG2

INT 21H

MOV AH, 2

MOV DL, 31H

INT 21H

MOV DL, BL

INT 21H

JMP EXIT

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**Lab-6**

1. **Copy string 1 to string 2 in reverse order**

ORG 100H

.DATA

STR1 DB "HELLO CLASS!$"

STR2 DB 15 DUP(?), '$'

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV ES, AX

STD

LEA SI, STR1+11

LEA DI, STR2

MOV CX, 11

MOVSB

REVERSE:

ADD DI, 2

MOVSB

LOOP REVERSE

DISPLAY:

MOV AH, 9

LEA DX, STR2

INT 21H

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

1. **Check if there is any vowel in a string.**

ORG 100H

.DATA

STR DB "HiLLL WYRLD$"

VOW DB "AEIOUaeiou$"

MSG1 DB "VOWEL FOUND$"

MSG2 DB "VOWEL NOT FOUND$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV ES, AX

CLD

MOV BX, 0

VOWEL:

CMP VOW[BX], '$'

JE DISPLAY\_N

LEA DI, STR

MOV AL, VOW[BX]

INC BX

MOV CX, 11

SCAN:

REPNE SCASB

JNE VOWEL

DISPLAY:

MOV AH, 9

LEA DX, MSG1

INT 21H

JMP EXIT

DISPLAY\_N:

MOV AH, 9

LEA DX, MSG2

INT 21H

JMP EXIT

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

1. **Check palindrome using CMPSB/CMPSW**

ORG 100H

.DATA

STR1 DB "HELLO CLASS!$"

STR2 DB 15 DUP(?), '$'

MSG1 DB "String is pallindrome!$"

MSG2 DB "String is not pallindrome!$"

MSG3 DB "ENTERED STRING=$"

MSG4 DB "REVERSE STRING=$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV ES, AX

STD

LEA SI, STR1+11

LEA DI, STR2

MOV CX, 11

MOVSB

REVERSE:

ADD DI, 2

MOVSB

LOOP REVERSE

DISPLAY:

MOV AH, 9

LEA DX, MSG3

INT 21H

MOV AH, 9

LEA DX, STR1

INT 21H

MOV AH,2

MOV DL,10

INT 21H

MOV DL,13

INT 21H

MOV AH, 9

LEA DX, MSG4

INT 21H

MOV AH, 9

LEA DX, STR2

INT 21H

MOV AH,2

MOV DL,10

INT 21H

MOV DL,13

INT 21H

COMPARE\_PALINDROME:

CLD

LEA SI,STR1

LEA DI,STR2

MOV CX,12

REPE CMPSB

JNE PRINT\_N

PRINT:

MOV AH, 9

LEA DX, MSG1

INT 21H

JMP EXIT

PRINT\_N:

MOV AH, 9

LEA DX, MSG2

INT 21H

JMP EXIT

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**Lab-7**

1. **Takes text as input and display text in reverse order when gets carriage return**

.MODEL SMALL

.STACK 100H

.DATA

PROMPT\_1 DB 'Enter the string : $'

PROMPT\_2 DB 0DH,0AH,'The string with words in reverse order : $'

COUNT DW 0

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

LEA DX, PROMPT\_1

MOV AH, 9

INT 21H

XOR CX, CX

MOV AH, 1

@INPUT:

INT 21H

CMP AL, 0DH

JE NL

PUSH AX

INC CX

JMP @INPUT

LEA DX, PROMPT\_2

MOV AH, 9

INT 21H

NL:

MOV AH,2

MOV DL,10

INT 21H

MOV DL,13

INT 21H

REVERSE:

MOV AH,2

POP DX

INT 21H

LOOP REVERSE

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

1. **Takes words separated by blanks, when gets carriage return displays text in reverse but same word order**

ORG 100H

.MODEL SMALL

.STACK 100H

.DATA

STR DB 50 DUP(?), '$'

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AH, 1

MOV CX, 0

LEA SI, STR

INPUT:

INT 21H

MOV BL, AL

CMP BL, 32

JE REVERSE

CMP BL, 13

JE REVERSE

PUSH BX

INC CX

JMP INPUT

REVERSE:

POP DX

MOV [SI], DL

INC SI

LOOP REVERSE

MOV [SI], 32

INC SI

CMP BL, 13

JE DISPLAY

JMP INPUT

DISPLAY:

MOV AH, 2

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

MOV AH, 9

LEA DX, STR

INT 21H

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**MID**

ORG 100H

.DATA

MSG1 DB "Enter a Hex digit: $"

MSG2 DB " The decimal value is: $"

MSG3 DB "Illegal character input.$"

MSG4 DB "Do you want to repeat the program?$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

INPUT:

MOV AH, 9

LEA DX, MSG1

INT 21H

MOV AH, 1

INT 21H

MOV BL, AL

NEWLINE:

MOV AH, 2

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

CHECK\_ATOF:

CMP BL, 41H

JB CHECK\_1TO9

CMP BL, 46H

JA CHECK\_a\_Tof

JMP CONVERT

CHECK\_1TO9:

CMP BL, 30H

JB ILLEGAL

CMP BL, 39H

JA ILLEGAL

MOV AH, 9

LEA DX, MSG2

INT 21H

MOV AH, 2

MOV DL, BL

INT 21H

JMP REPEAT

CHECK\_a\_Tof:

CMP BL, 61H

JB ILLEGAL

CMP BL, 66H

JA ILLEGAL

JMP CONVERT\_LOWERCASE

CONVERT:

SUB BL, 11H

MOV AH, 9

LEA DX, MSG2

INT 21H

MOV AH, 2

MOV DL, 31H

INT 21H

MOV DL, BL

INT 21H

JMP REPEAT

CONVERT\_LOWERCASE:

SUB BL, 31H

MOV AH, 9

LEA DX, MSG2

INT 21H

MOV AH, 2

MOV DL, 31H

INT 21H

MOV DL, BL

INT 21H

JMP REPEAT

ILLEGAL:

LEA DX, MSG3

MOV AH, 9

INT 21H

MOV AH, 4CH

INT 21H

REPEAT:

MOV AH, 2

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

LEA DX, MSG4

MOV AH, 9

INT 21H

MOV AH, 1

INT 21H

MOV CL, AL

CMP CL, 'Y'

MOV AH, 2

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

JE INPUT

CMP CL, 'y'

JE INPUT

JMP EXIT

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**PALINDROME**

ORG 100H

.DATA

STR DB 15 DUP(?), '$'

REV DB 15 DUP(?), '$'

MSG1 DB "Enter String: $"

MSG2 DB "It is a palindrome$"

MSG3 DB "It is not a palindrome$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AH, 9

LEA DX, MSG1

INT 21H

MOV BX, 0

MOV AH, 1

INPUT:

INT 21H

CMP AL, 13

JE REVERSE\_CODE

MOV STR[BX], AL

INC BX

JMP INPUT

REVERSE\_CODE:

LEA SI, REV

MOV CX, BX

DEC BX

REVERSE:

MOV AL, STR[BX]

MOV [SI], AL

INC SI

DEC BX

LOOP REVERSE

COMPARE\_CODE:

MOV BX, 0

XOR SI, SI

LEA SI, STR

COMPARE:

MOV CL, [SI]

MOV DL, REV[BX

INC SI

INC BX

CMP CL, '$'

JE PRINT

CMP CL, DL

JE COMPARE

PRINT\_NEG:

MOV AH, 2

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

MOV AH, 9

LEA DX, MSG3

INT 21H

JMP EXIT

PRINT:

MOV AH, 2

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

MOV AH, 9

LEA DX, MSG2

INT 21H

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**EVEN OR ODD**

ORG 100H

.DATA

ARR DB 21, 32, 1, 22, 23, 45, 78, 42, 9

MSG1 DB "Even Number: $"

MSG2 DB "Odd Number $"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV BH, 0

MOV BL, 0

MOV CX, 9

LEA SI, ARR

CHECK:

MOV AL, [SI]

TEST AL, 1

JZ COUNT\_E

JMP COUNT\_O

COUNT\_E:

INC BL

JMP LP

COUNT\_O:

INC BH

JMP LP

LP:

INC SI

LOOP CHECK

PRINT:

MOV AH, 9

LEA DX, MSG2

INT 21H

MOV AH, 2

MOV DL, BH

ADD DL, 30H

INT 21H

MOV DL, 10

INT 21H

MOV DL, 13

INT 21H

MOV AH, 9

LEA DX, MSG1

INT 21H

MOV AH, 2

MOV DL, BL

ADD DL, 30H

INT 21H

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**QUIZ-2 Solution**

ORG 100H

.DATA

STR DB "HELLO WORLD$"

VOW DB "AEIOUaeiou$"

MSG1 DB "VOWELS FOUND$"

MSG2 DB "VOWEL NOT FOUND$"

MSG3 DB "VOWEL FOUND$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV ES, AX

CLD

LEA SI, VOW

;MOV BX, 0

XOR BX, BX

VOWEL:

CMP [SI], '$'

;JE DISPLAY\_N

JE DISPLAY

LEA DI, STR

MOV AL, [SI]

INC SI

MOV CX, 11

SCAN:

REPNE SCASB

JNE VOWEL

INC BL ;increment bl if equal

JMP VOWEL

DISPLAY:

CMP BL, 0

JE DISPLAY\_N ;no vowel found

CMP BL, 1

JE DISPLAY\_1 ;1 vowel found

;below is the code for more than one vowels found

MOV AH, 2

MOV DL, BL

ADD DL, 30H

INT 21H

MOV DL, 32

INT 21H

MOV AH, 9

LEA DX, MSG1

INT 21H

JMP EXIT

DISPLAY\_N: ;no vowel

MOV AH, 9

LEA DX, MSG2

INT 21H

JMP EXIT

DISPLAY\_1: ;1 vowel

MOV AH, 2

MOV DL, 31

INT 21H

MOV DL, 32

INT 21H

MOV AH, 9

LEA DX, MSG3

INT 21H

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN