

## Dubai International Academic City

**CS/ECE/INSTR/EEE F241**  
**MICROPROCESSORS AND INTERFACING**  
**LABORATORY MANUAL**  
**II Semester 2021-22**

**EXPERIMENT-4**  
**Code Conversions/ Display Programs**

**Code Conversions**

**P1. Write a program to convert 4-digit BCD number to HEXADECIMAL number and store the result in memory.**

**Handwritten codes:**

2021A7PS0136U - K.Yashwanth - P1 - Code Conversions

```

MODEL SMALL
Stack 20
.DATA
ORG 1000H
BCD DW 1234H
HEX DW 0

.CODE
START:
MOV AX, @DATA
MOV DS, AX
MOV AX, 0000H
MOV BX, 0001H
CALL BCD2BIN
MOV BX, 000AH
CALL BCD2BIN
MOV BX, 03E9H
CALL BCD2BIN
INT 3
BCD2BIN PROC NEAR
MOV AX, BCD
AND AX, 000FH
MUL BX
ADD HEX, AX
MOV CL, 04
ROR BCD, CL
RET
BCD2BIN ENDP
END START

```

**Solution Screenshot:**

-&gt; Hex no. (output)

-&gt; BCD no. (input)

Lower byte	Higher byte
ds : 1004 34	12
D2	04
00 00 23 41	41
41	#A
ds : 100C 3A	10
00 00 00	00
03 03 00 00	:>
00 00 E0 48	**
DF 01 C5 15	H B S
ds : 101C E8 03 21	AD 48 16 32
00 00 00	32
AD 48 16 32	32
32	!H 2

Lower byte

Higher byte

**P2.** Write a program to convert the given HEXADECIMAL digit to ASCII byte and store the result in memory.

**Handwritten codes:**

2021A7PS0136U - K.Yashwanth - P2

```

.MODEL SMALL
.STACK 20
.DATA
ORG 1000H
Hex-Digit DB 38H
ASCII DB ?

.CODE
START:
MOV AX, @DATA
MOV DS, AX
MOV AL, Hex-Digit
CMP AL, 3AH
JC SUB30
SUB AL, 071H
SUB30:
SUB AL, 30H
MOV ASCII, AL
INT 3
END START

```

**Solution Screenshot:**

-> Hex no. (output)      -> Decimal no. (input)

```

ds:1004 38 08 00 00 00 00 00 00 00 80
ds:100C 00 00 00 00 03 48 00 00 00 H
ds:1014 00 00 AE 48 DF 01 C5 15 «H»+§
ds:101C 08 48 14 00 AD 48 02 32 •H¶ iH§2

```

**Display Programs**

P3. Write a program to display the hexadecimal byte 45H on the screen using DOS interrupts. Previous program is HEXASC (HEX TO ASCII), refer for the HEXASC procedure.

**Handwritten codes:**

2021A7PS0136U - K.Yashwanth - P3 - Display Programs

```

.MODEL SMALL
.STACK 20
.CODE
START:
    MOV AL, 45H
    MOV BL, AL
    AND AL, 0F0H
    ROR AL, 4
    CALL HEXASC
    MOV DL, AL
    MOV AH, 02
    INT 21H
    MOV AL, BL
    AND AL, 0F1H
    CALL HEXASC
    MOV DL, AL
    MOV DL, AL
    MOV AH, 02
    INT 21H
    MOV AH, 4CH
    INT 21H
    HEXASC!
    CMP AL, 0AH
    JB NUM
    ADD AL, 07
    NUM!
    ADD AL, 30H
    RET
END START

```

**Solution Screenshot:**

C:\TASM>LAB4P3  
45

Displayed hex. no.

**P4. Write a program to input two single-digit hex numbers from keyboard and display their product on the screen.**

**Handwritten codes:**

2021A7PS0136U - K. Yashwanth - P4

```

.MODEL SMALL
.STACK 20
.CODE
.START:
    CALL READKB
    MOV BL, AL
    CALL NXTLINE
    CALL READKB

    MUL BL
    MOV BL, AL
    CALL NXTLINE
    CALL DISP

    MOV AH, 0CH
    INT 21H

    READKB PROC NEAR
        MOV AH, 01
        INT 21H
        CALL ASCHEH
        RET
    READKB ENDP

    ASCHEX PROC NEAR
        CMP AL, 3AH
        JC SUB30
        SUB AL, 07H
        SUB AL, 30H
        SUB AL, 30H
        SUB AL, 0FH
        RET
    ASCHEX ENDP

    NXTLINE PROC NEAR

```

```

MOV AH, 02H
MOV DL, 0AH
INT 21H
RET
NEXTLINE ENDP
DISP PROC NEAR
MOV AL, BL
AND AL, 0FH
ROR AL, 4
CALL HEXASC
MOV DL, AL
MOV AH, 02H
INT 21H
MOV AL, BL
AND AL, 0FH
CALL HEXASC
MOV DL, AL
MOV AH, 02H
INT 21H
RET
DISP ENDP
HEXASC PROC NEAR
CMP AL, 0AH
JB NUM
ADD AL, 07H
NUM:
ADD AL, 030H
RET
HEXASC ENDP
ENDS START

```

### Solution Screenshot:

The screenshot shows a terminal window with the following text:

```
C:\TASM>LAB4P4
3
4
0C
C:\TASM>LAB4P4
A
C
78
```

Annotations with yellow arrows point to specific parts of the output:

- An arrow points from the number "3" to the text "1st input".
- An arrow points from the number "4" to the text "2nd input".
- An arrow points from the number "78" to the text "Product (result)".

**Exercise Programs:**

**Q2. Write a program to accept a character from keyboard and display its ASCII equivalent value on the screen.**

**Handwritten codes:**

2021A7PS0136U - K.Yashwanth - Exercise 02

```
.MODEL SMALL
.STACK 20
.CODE
.START:
    CALL READKB
    MOV BL, AL
    CALL NEXTLINE
    CALL DISP
    MOV AH, 4CH
    INT 21H

    READKB PROC NEAR
    MOV AH, 01
    INT 21H
    RET
    READKB ENDP

    NEXTLINE PROC NEAR
    MOV AH, 02
    MOV DL, 0AH
    INT 21H
    MOV DL, 0DH
    INT 21H
    RET
    NEXTLINE ENDP

    DISP PROC NEAR
    MOV AL, BL
    AND AL, 0F0H
    ROR AL, 4
    CALL HEXASC
    MOV DL, AL
    MOV AH, 02
    INT 21H
    MOV AL, BL
    AND AL, 0FH
```

```
CALL HEXASC
MOV DL, AL
MOV AH, 02
INT 21H
RET
DISP ENDP

HEXASC PROC NEAR
CMP AL, 0AH
JB NUM
ADD AL, 07
NUM!
ADD AL, 30H
RET
HEXASC ENDP
END START
```

### Solution Screenshot:

```
C:\TASM>LAB4A1
4
34
C:\TASM>LAB4A1
A
41
C:\TASM>LAB4A1
B
42
C:\TASM>LAB4A1
9
39
```

Keyboard input

Ascii output