NAME: GINI CHACKO

SEMESTER: IV

CLASS: SE COMPS B

BATCH: B

ROLL: 8942

TOPIC: MP EXPERIMENT 5:

To Perform Code Conversion

- Hex to BCD
- BCD to Hex
- ASCII to BCD
- BCD to ASCII

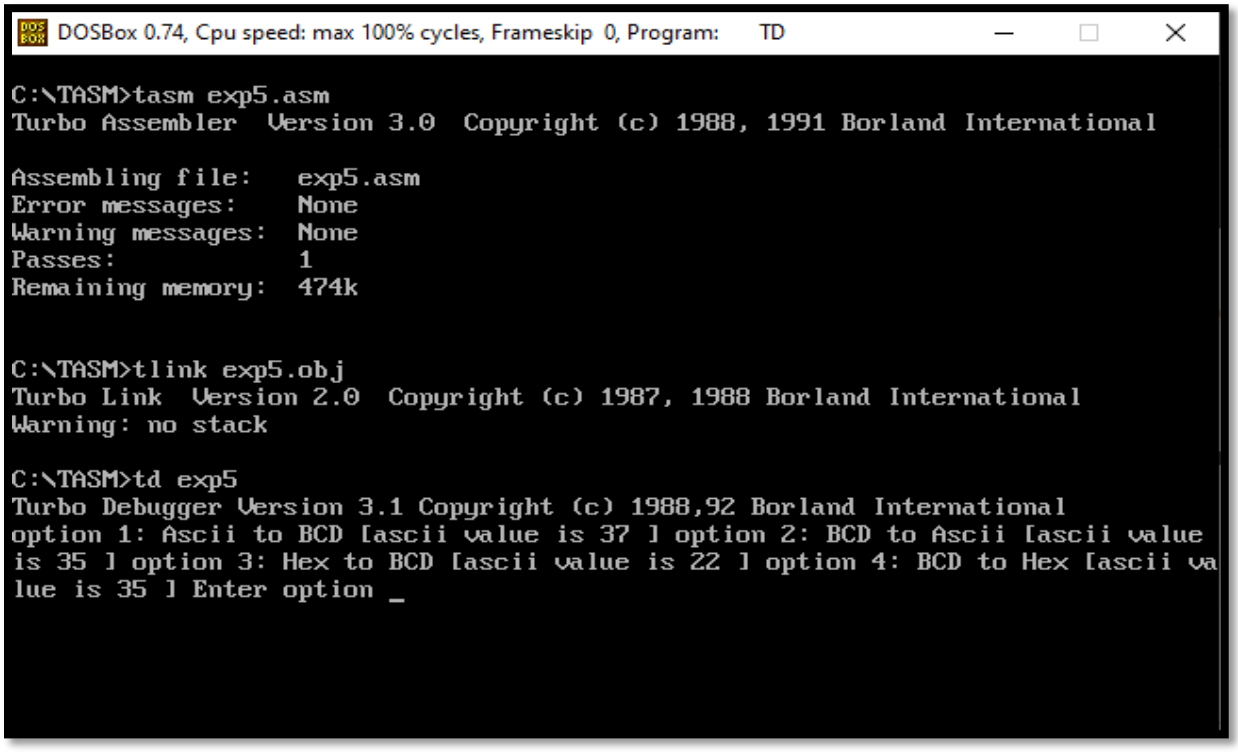
CODE:

```
.8086
.model  small
.data
num   db  37h
num1  db 35h
num2  db 22h
num3  db  35h

res  db  ?
res1  db  ?
res2 db  ?
result   db  ?

msg2  db  'option 1: Ascii  to BCD  [ascii  value is  37 ] $'
msg3  db  'option 2: BCD  to Ascii  [ascii  value is  35 ] $'
msg4 db  'option 3: Hex  to BCD  [ascii  value is  22 ] $'
msg5  db  'option 4: BCD  to Hex  [ascii  value is  35 ] $'
msg1 db  'Enter  option  $'

.code
Start:
mov  ax,@data
mov  ds,  ax

lea dx,msg2
mov  ah,09h
int 21h

lea dx,msg3
mov  ah,09h
int 21h

lea dx,msg4
mov  ah,09h
int 21h

lea dx,msg5
mov  ah,09h
```

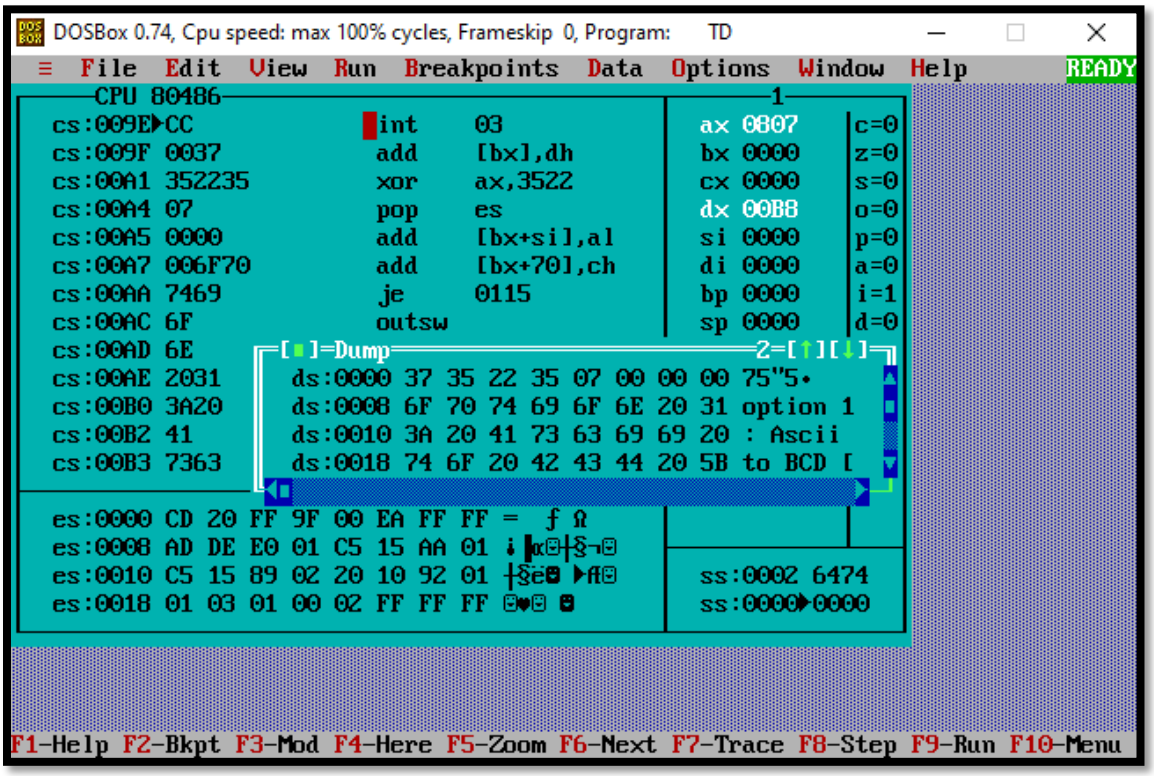
```
int 21h

lea dx,msg1
mov  ah,09h
int 21h

mov  ah,08h
int  21h

cmp   al,31h
jnz   next
;AsciiToBCD
        mov al, num
        sub al, 30h
        mov res ,al
        jmp exit
next :cmp  al,32h
jnz  next1

;BCDtoAscii
        mov al,num1
        and al,0F0h
        ror al, 4
        add al,30h
        mov bl,num1
        and bl,0Fh
        add bl,30h
        mov res1, al
        mov res2, bl
        jmp exit
next1:cmp  al,33h
jnz next2

;HextoDEC
        mov al,num2
        mov ah,00h
        mov bl,0Ah
        div bl
        ror al,04h
        add al,ah
```

```
        jmp exit

next2:cmp  al,34h
;BCDtoHEX
        mov al,num3
        mov dl,00h
        mov bl,al
        and bl, 0F0h
        mov cl,04h
        ror bl,cl
        back: add dl, 0Ah
              dec bl
        jnz  back
        and al,0Fh
        add dl,  al
        mov result ,dl
        jmp exit

exit:int  3h
end  start
```

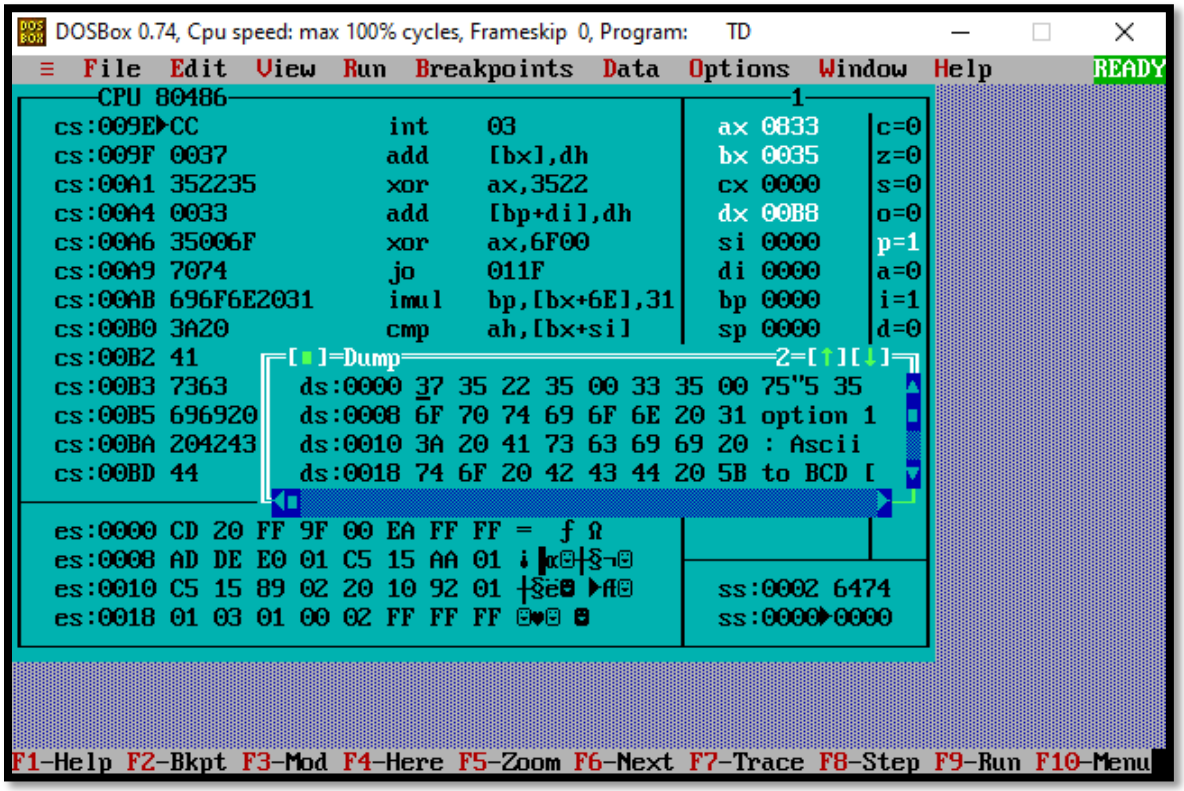
OUTPUT:



1.] ASCII to BCD (Option 1)

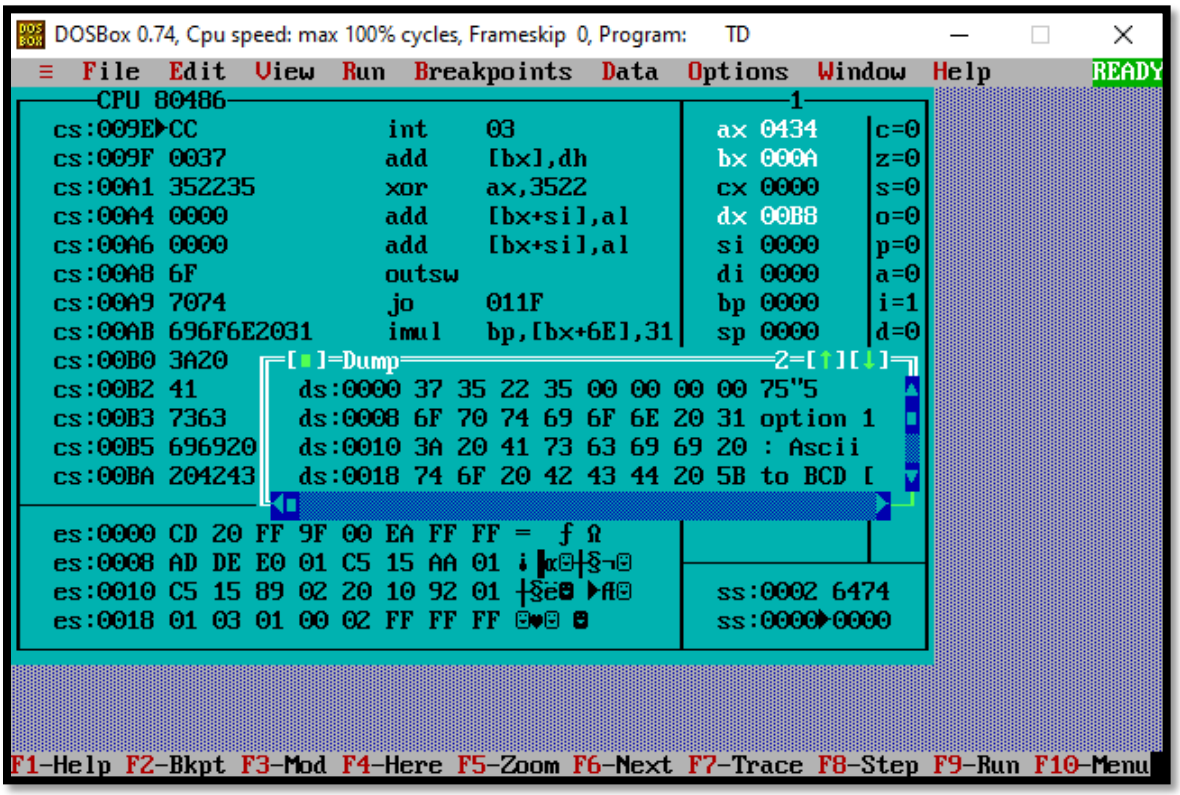


2.] BCD to ASCII (Option 2)

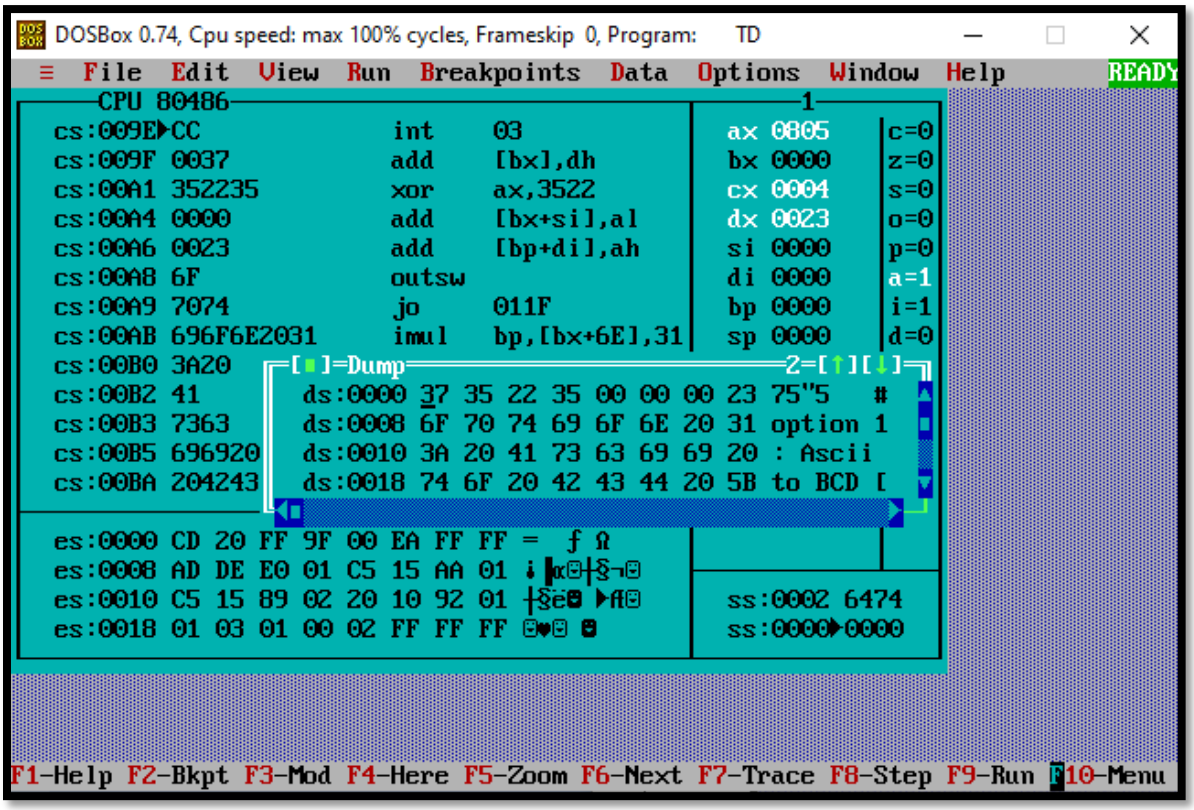


GINI CHACKO 8942

3.] Hex to BCD (Option 3)



4.] BCD to Hex (Option 4)



POSTLAB QUESTIONS:

1. Write any 10 Bit manipulation instructions with example
Ans:

These instructions are used to perform operations where data bits are involved, i.e. operations like logical, shift, etc.

Following is the list of instructions under this group −

- **Instructions to perform logical operation**

- **NOT** − Used to invert each bit of a byte or word.
- **AND** − Used for adding each bit in a byte/word with the corresponding bit in another byte/word.
- **OR** − Used to multiply each bit in a byte/word with the corresponding bit in another byte/word.
- **XOR** − Used to perform Exclusive-OR operation over each bit in a byte/word with the corresponding bit in another byte/word.
- **TEST** − Used to add operands to update flags, without affecting operands.

- **Instructions to perform shift operations**

- **SHL/SAL** − Used to shift bits of a byte/word towards left and put zero(S) in LSBs.
- **SHR** − Used to shift bits of a byte/word towards the right and put zero(S) in MSBs.
- **SAR** − Used to shift bits of a byte/word towards the right and copy the old MSB into the new MSB.

- **Instructions to perform rotate operations**

- **ROL** − Used to rotate bits of byte/word towards the left, i.e. MSB to LSB and to Carry Flag [CF].
- **ROR** − Used to rotate bits of byte/word towards the right, i.e. LSB to MSB and to Carry Flag [CF].

- **RCR** − Used to rotate bits of byte/word towards the right, i.e. LSB to CF and CF to MSB.
- **RCL** − Used to rotate bits of byte/word towards the left, i.e. MSB to CF and CF to LSB.

2. Explain 8086 in Minimum and Maximum mode.

Ans:

MINIMUM MODE	MAXIMUM MODE
1.In minimum mode there can be only one processor i.e. 8086.	1.In maximum mode there can be multiple processors with 8086, like 8087 and 8089.
2. MN/MX is 1 to indicate minimum mode.	2. MN/MX is 0 to indicate maximum mode.
3. ALE for the latch is given by 8086 as it is the only processor in the circuit.	3. ALE for the latch is given by 8288 bus controller as there can be multiple processors in the circuit.
4. DEN and DT/R for the trans-receivers are given by 8086 itself.	4. DT/R for the trans-receivers are given by 8288 bus controller.
5. Direct control signals M/IO, RD and WR are given by 8086.	5. Instead of control signals, each processor generates status signals called S2, S1 and S0.
6. Control signals M/IO, RD and WR are decoded by a 3:8 decoder like 74138	6. Status signals S2,S1 and S0 are decoded by a bus controller like 8288 to produce control signals.
7. INTA is given by 8086 in response to an interrupt on INTR line.	7. INTA is given by 8288 bus controller in response to an interrupt on INTR line.
8. HOLD and HLDA signals are used for bus request with a DMA controller like 8237.	8. RQ/GT lines are used for bus requests by other processors like 8087 or 8089.
9. The circuit is simpler.	9.The circuit is more complex.
10. Multiprocessing cannot be performed hence performance is lower.	10. As multiprocessing can be performed, it can give very high performance.