Code:-

```
-------------------------------------------------------------------------
%macro display 2
mov rax,1
mov rdi,1
mov rsi,%1
mov rdx,%2
syscall
%endm

%macro input 2
mov rax,0
mov rdi,0
mov rsi,%1
mov rdx,%2
syscall
%endm

global _start

section .data                                   ;Taking input from user

msg db "enter BCD no:"
msg_len equ $-msg
msg1 db 10,13,"BCD equivalent for given hex num is:"
msg1_len equ $-msg1
msg2 db 10,13,"Hex equivalent for given BCD is:"
msg2_len equ $-msg2
num dw 1,000Ah,64h,3E8h,2710h
msg3 db 10,13,"enter hex num:"
msg3_len equ $-msg3
msg4 db "enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:"
msg4_len equ $-msg4
msg5 db 10,13,"----------error-------",10,13
msg5_len equ $-msg5

section .bss
bcd_no resb 6
```

```
hex_con resb 4
rem_dx resb 5
rem_dx_ascii resb 5
hex_num_ascii resb 4
hex_num_in resb 5
ch_num resb 2

section .text
_start:

takech: display msg4,msg4_len
input ch_num,2
cmp byte[ch_num],31h
jne dnhtb


call BCD_input
call bcdtohex
call disphexnum

dnhtb:
cmp byte[ch_num],32h
jne dnext

call hexinput
call hextobcd
call dispbcdnum

dnext:
cmp byte[ch_num],33h
je ext
display msg5,msg5_len
jmp takech


ext: mov rax,60
mov rdi,0
syscall
```

```
BCD_input:
display msg,msg_len
input bcd_no,6
mov rsi,bcd_no
mov cl,05
mov ch,30h
up: sub [rsi],ch
inc rsi
dec cl
jnz up
ret


bcdtohex:
mov word[hex_con],0000h
mov rdi,num
dec rsi
mov cl,05
up1: mov ah,00
mov al,[rsi]
mov bx,[rdi]
mul bx
add [hex_con],ax
dec rsi
inc rdi
inc rdi
dec cl
jnz up1

ret


disphexnum:

mov rdi,hex_num_ascii
mov cl,4
mov ax,[hex_con]
updhn: rol ax,4                         ;RAX= 2345678123456781
mov bx,ax
and al,0fh                          ;RAX= 0000000000000001
```

```
cmp al,09
ja dn2
add al,30h
jmp dn3
dn2: add al,37h
dn3: mov [rdi],al
inc rdi
mov ax,bx
dec cl
jnz updhn
display msg2,msg2_len
display hex_num_ascii,4


ret

hextobcd:
mov rsi,rem_dx
mov cl,05

mov bx,0Ah
uphex: mov dx,00
div bx
mov [rsi],dl
inc rsi
dec cl
jnz uphex

ret

dispbcdnum:
dec rsi
mov rdi,rem_dx_ascii
mov cl,05
uphex1: add byte[rsi],30h
mov al,[rsi]
mov [rdi],al
dec rsi
inc rdi
dec cl
```

```
jnz uphex1
display msg1,msg1_len
display rem_dx_ascii,5
ret


hexinput:
display msg3,msg3_len
input hex_num_in,5
mov cl,04
mov rsi,hex_num_in
uphex2: mov al,[rsi]
cmp al,39h
ja dnhex
sub al,30h
jmp nxt
dnhex:
sub al,37h
nxt:
mov [rsi],al
inc rsi
dec cl
jnz uphex2

mov cl,04
mov rsi,hex_num_in
mov ax,0000
uphex3: mov dl,[rsi]
rol ax,4
add al,dl

inc rsi
dec cl
jnz uphex3

re
```

Output:-

Code Input Output Run Save

```
[Your program generated a lot of data and we will show the first 10 KB of output. Please check if your program has infinite loops or
if it is an interactive program, you may need to add input via the Input tab.]

enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:43B0
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:2060
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:FD10
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:D9C0
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:B670
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:FA20
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:D6D0
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:B380
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
Hex equivalent for given BCD is:9030
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:enter BCD no:
```

Code Input Output Run Save

```
1 2
```

Code Input Output Run Save

```
1 1
```

Code Input Output Run Save

```
[Your program generated a lot of data and we will show the first 10 KB of output. Please check if your program has infinite loops or
if it is an interactive program, you may need to add input via the Input tab.]

enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:
enter hex num:
BCD equivalent for given hex num is:56797
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:
enter hex num:
BCD equivalent for given hex num is:08754
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:
enter hex num:
BCD equivalent for given hex num is:34952
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:
enter hex num:
BCD equivalent for given hex num is:56797
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:
enter hex num:
BCD equivalent for given hex num is:43706
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:
enter hex num:
BCD equivalent for given hex num is:00016
----------error-------
enter your choice for 1. BCD to HEX 2. HEX to BCD 3.EXIT:
enter hex num:
BCD equivalent for given hex num is:21861
----------error-------
```

Nikhil Vinod Khodake
9022
SE Computer
MP Assignments