```
section .data

      menumsg db 10,10,'###### Menu for Code Conversion ######'
            db 10,'1: Hex to BCD'
            db 10,'2: BCD to Hex'
            db 10,'3: Exit'
            db 10,10,'Please Enter Choice::'
      menumsg_len equ $-menumsg

      hexinmsg db 10,10,'Please enter 4 digit hex number::'
      hexinmsg_len equ $-hexinmsg

      bcdopmsg db 10,10,'BCD Equivalent::'
      bcdopmsg_len equ $-bcdopmsg

      bcdinmsg db 10,10,'Please enter 5 digit BCD number::'
      bcdinmsg_len equ $-bcdinmsg

      hexopmsg db 10,10,'Hex Equivalent::'
      hexopmsg_len equ $-hexopmsg

section .bss
      numascii resb 06  ;common buffer for choice, hex and bcd input
      outputbuff resb 02
      dispbuff resb 08

      %macro display 2
      mov rax,01
      mov rdi,01
      mov rsi,%1
      mov rdx,%2
      syscall
      %endmacro

      %macro accept 2
      mov rax,0
      mov rdi,0
      mov rsi,%1
      mov rdx,%2
      syscall
      %endmacro

section .text

global _start

_start:

menu:
      display menumsg,menumsg_len
      accept numascii,2

      cmp byte [numascii],'1'
      je hex2bcd_proc

      cmp byte [numascii],'2'
      je bcd2hex_proc

      cmp byte [numascii],'3'
      je exit
      jmp _start

exit:
      mov rax,60
      mov rbx,0
      syscall

hex2bcd_proc:
      display hexinmsg,hexinmsg_len
      accept numascii,5
      call packnum
      mov ax,bx
      mov rcx,0
      mov bx,10   ;Base of Decimal No. system
h2bup1:
      mov dx,0
      div bx
      push rdx
      inc rcx
      cmp ax,0
      jne h2bup1
      mov rdi,outputbuff
h2bup2:
      pop rdx
      add dl,30h
      mov [rdi],dl
      inc rdi
      loop h2bup2

      display bcdopmsg,bcdopmsg_len
      display outputbuff,5
      jmp menu

bcd2hex_proc:
      display bcdinmsg,bcdinmsg_len
      accept numascii,6

      display hexopmsg,hexopmsg_len
      mov rsi,numascii
      mov rcx,05
      mov rax,0
      mov ebx,0ah
b2hup1:
      mov rdx,0
      mul ebx
      mov dl,[rsi]
      sub dl,30h
      add rax,rdx
      inc rsi
      loop b2hup1
      mov ebx,eax
      call disp32_num
      jmp menu
packnum:
      mov bx,0
      mov ecx,04
      mov esi,numascii
up1:
      rol bx,04
      mov al,[esi]
      cmp al,39h
      jbe skip1
      sub al,07h
skip1:
      sub al,30h
      add bl,al
      inc esi
      loop up1
      ret
disp32_num:
      mov rdi,dispbuff  ;point esi to buffer
      mov rcx,08        ;load number of digits to display
dispup1:
      rol ebx,4   ;rotate number left by four bits
      mov dl,bl   ;move lower byte in dl
      and dl,0fh  ;mask upper digit of byte in dl
      add dl,30h  ;add 30h to calculate ASCII code
      cmp dl,39h  ;compare with 39h
      jbe dispskip1     ;if less than 39h akip adding 07 more
      add dl,07h  ;else add 07
dispskip1:
      mov [rdi],dl      ;store ASCII code in buffer
      inc rdi           ;point to next byte
      loop dispup1      ;decrement the count of digits to display
                  ;if not zero jump to repeat

      display dispbuff+3,5    ;Dispays only lower 5 digits as upper three are '0'
      ret
```