

```

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

%macro read 2
    mov rax,0
    mov rdi,0
    mov rsi,%1
    mov rdx,%2
    syscall
%endmacro
section .data
    msg1 db 0x0A,'1: Hex to BCD',0x0A,'2: BCD to Hex',0x0A,'3: Exit',0x0A
    len1 equ $-msg1

    msg2 db 10,10,'Enter 4 digit HEX number - '
    len2 equ $-msg2

    msg3 db 10,10,'BCD Equivalent - '
    len3 equ $-msg3

    msg4 db 10,10,'Enter 5 digit BCD number - '
    len4 equ $-msg4

    msg5 db 10,10,'HEX Equivalent - '
    len5 equ $-msg5

section .bss

    noascii resb 06
    opbuff resb 02
    buff resb 08

section .text

    global _start
_start:

menu:
    write msg1,len1
    read noascii,2

    cmp byte [noascii],'1'
    je HtoB

    cmp byte [noascii],'2'
    je BtoH

    cmp byte [noascii],'3'
    je exit

```

```

    jmp _start

exit:
    mov rax,60
    mov rbx,0
    syscall

HtoB:
    write msg2,len2
    read noascii,5
    call packnum
    mov ax,bx
    mov rcx,0
    mov bx,10
h2bup1:    mov dx,0
    div bx
    push rdx
    inc rcx
    cmp ax,0
    jne h2bup1
    mov rdi,opbuff

h2bup2:    pop rdx
    add dl,30h
    mov [rdi],dl
    inc rdi
    loop h2bup2

    write msg3,len3
    write opbuff,5
    jmp menu

BtoH:
    write msg4,len4
    read noascii,6

    write msg5,len5

    mov rsi,noascii
    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,noascii
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,buff
    mov rcx,08

```

```

dispup1:
    rol ebx,4
    mov dl,bl
    and dl,0fh
    add dl,30h
    cmp dl,39h
    jbe dispskip1
    add dl,07h

```

```

dispskip1:
    mov [rdi],dl
    inc rdi
    loop dispup1

```

```

    write buff+3,5

```

```

    ret

```