Assignment2.asm

```
%macro WRITE 02
mov rax,01
mov rdi,01
mov rsi,%1
mov rdx,%2
syscall
%endmacro

%macro READ 02
mov rax,00
mov rdi,00
mov rsi,%1
mov rdx,%2
syscall
%endmacro

section .data

msg1 db "Enter BCD no ", 10
len1 equ $-msg1

msg2 db "Hex Equivalent", 10
len2 equ $-msg2

msg3 db "Enter HEX no ", 10
len3 equ $-msg3

msg4 db "BCD equivalent", 10
len4 equ $-msg4

menu db 10,"1.BCD to HEX", 10
db "2.HEX to BCD", 10
db "3.exit", 10
db "Enter your choice" , 10
menulen equ $-menu

msg5 db "Wrong Choice", 10
len5 equ $-msg5

section .bss
char_buff resb 17
actl resq 1
ans resq 1
cnt resb 1
x resb 1
choice resb 2

section .text
global _start
_start: WRITE menu, menulen
READ choice, 2
```

```
cmp byte[choice], 31H
je bcdtohex
cmp byte[choice], 32H
je hextobcd
cmp byte[choice], 33H
je exit
WRITE msg5, len5
jmp _start

bcdtohex: WRITE msg1, len1

READ char_buff, 17
dec rax
mov [actl], rax

mov rax, 00H
mov rsi, char_buff
mov rbx, 0AH
up: mul rbx
mov rdx, 00H
mov dl, byte[rsi]
sub dl, 30H
add rax, rdx
inc rsi
dec qword[actl]
jnz up

mov[ans], rax
WRITE msg2, len2
hi:mov rbx,[ans]
call display

jmp _start

hextobcd: WRITE msg3, len3
READ char_buff, 17
call accept
mov byte[cnt], 00
mov rax, rbx
up1: mov rdx, 00H
mov rbx, 0AH
div rbx
push rdx
inc byte[cnt]
cmp rax, 00
jne up1
WRITE msg4, len4
up2: pop rdx
add dl, 30H
mov byte[x],dl
WRITE x, 01
dec byte[cnt]
jnz up2

jmp _start
```

```
exit:mov rax, 60
mov rdi, 00
syscall


display:
mov rsi, char_buff
mov rcx, 16
above: rol rbx, 04H
mov dl, bl
and dl, 0FH
cmp dl, 09H
jbe add30
add dl,07H
add30 : add dl, 30H
mov byte [rsi], dl
inc rsi
dec rcx
jnz above
WRITE char_buff, 17
RET

accept: dec rax
mov [actl],rax
mov rbx, 00H
mov rsi, char_buff
up4:shl rbx, 04H
mov rdx, 00H
mov dl, byte[rsi]

cmp dl, 39H
jbe sub30
sub dl, 07H
sub30: sub dl, 30H

add rbx, rdx
inc rsi
dec qword[actl]
jnz up4
ret
```