

## SC165

### Assignment No- 6

Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. (Wherever necessary, use 64-bit registers).

```
%macro write 2
mov rax,1
mov rdi,1
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 .data
msg1 db 10,'1.Hex to BCD',10
      db '2.BCD to Hex',10
      db '3.Exit',10
      db 'Enter choice: ',10
len1 equ $-msg1
msg2 db "Wrong choice.",10
len2 equ $-msg2
msg3 db "Enter 4 digit Hex: ",10
len3 equ $-msg3
msg4 db "Equivalent BCD: ",10
len4 equ $-msg4
nw db 10
msg5 db "Enter 5 digit BCD: ",10
len5 equ $-msg5
msg6 db "Equivalent Hex: ",10
len6 equ $-msg6
```

```
section .bss
```

```
numascii resb 6
result resb 6
opbuff resb 5
```

```
section .text
global _start
_start:
write msg1,len1
accept numascii,2
```

```
case1: cmp byte[numascii],'1'
jne case2
call h2b
jmp _start
```

```
case2: cmp byte[numascii],'2'
jne case3
call b2h
jmp _start
```

```
case3: cmp byte[numascii],'3'
jne case4
mov rax,60
mov rdi,0
syscall
```

```
case4: write msg2,len2
jmp _start
```

```
h2b:
write msg3,len3
accept numascii, 5
call packnum
mov rcx,0
mov ax,bx
mov bx,10
```

```
up1: mov rdx,0
div bx
push rdx
inc rcx
cmp ax,0
jne up1
mov rdi,opbuff
```

```
up2: pop rdx
add dl,30h
mov [edi],dl
inc rdi
loop up2
write msg4,len4
write opbuff,5
write nw,1
```

```
ret
```

```
b2h:
write msg5,len5
accept numascii,6
write msg6,len6
mov rsi,numascii
mov rcx,05
mov eax,0
mov ebx,10
```

```
l1:
mov edx,0
mul ebx
mov dl,[esi]
sub dl,30h
add eax,edx
inc esi
loop l1
call display
```

```
ret
packnum:
mov bx,0
mov ecx,04
mov esi,numascii
```

```
up: rol bx,04
mov al,[esi]
cmp al,39h
jbe skip1
sub al,07h
```

```
skip1: sub al,30h
```

```

add bl,al
inc esi
loop up
ret
display:
mov ebx,eax
mov cx,4
mov edi,result
up3:
rol bx,4
mov al,bl
and al,0fh
cmp al,09h
jg add_37
add al,30h
jmp skip
add_37:
add al,37h

skip:
mov [edi],al
inc edi
dec cx
jnz up3
write result,4
ret

```

[OBJ]

The screenshot shows a terminal window with the following content:

```

hardware@hardware-System-Product-Name: ~/Desktop
hardware@hardware-System-Product-Name:~$ cd Desktop
hardware@hardware-System-Product-Name:~/Desktop$ nasm -f elf64 d.asm
hardware@hardware-System-Product-Name:~/Desktop$ ld -o d d.o
hardware@hardware-System-Product-Name:~/Desktop$ ./d

1.HEX to BCD
2.BCD to HEX
3.EXIT
Enter your Choice:1
000F
15
1.HEX to BCD
2.BCD to HEX
3.EXIT
Enter your Choice:2
You are in B2H
00010
000A
1.HEX to BCD
2.BCD to HEX
3.EXIT
Enter your Choice:3
hardware@hardware-System-Product-Name:~/Desktop$

```