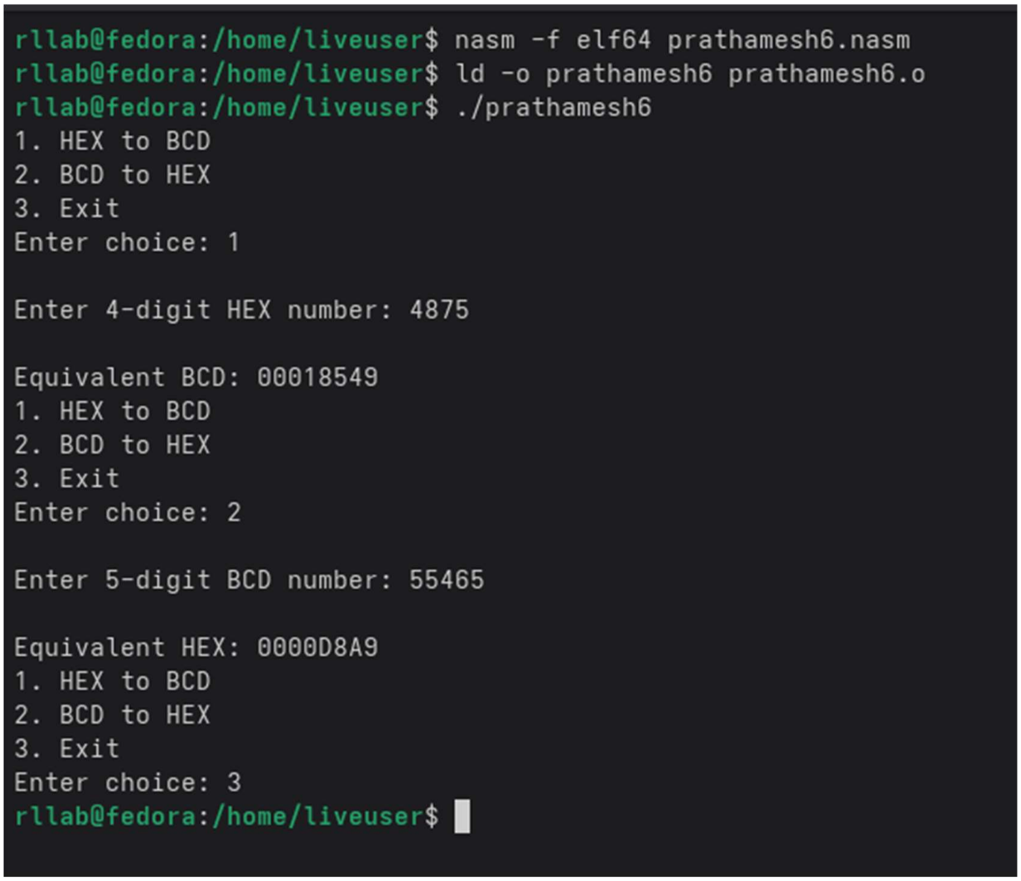
Practical6

```
;HEX<->BCD conversion
Section .data
msg: db 0x0A
len: equ $-msg
msg1: db "1. HEX to BCD",0x0A
db "2. BCD to HEX",0x0A
db "3. Exit",0x0A
db "Enter choice: "
len1: equ $-msg1
msg2: db 10,"Enter 4-digit HEX number: "
len2: equ $-msg2
msg3: db 10,"Equivalent BCD: "
len3: equ $-msg3
msg4: db 10,"Enter 5-digit BCD number: "
len4: equ $-msg4
msg5: db 10,"Equivalent HEX: "
len5: equ $-msg5
msg6: db 10,"Invalid choice!",0x0A
len6: equ $-msg6
;***************************************************
Section .bss
chc: resb 0x01
temp: resb 0x06
data: resd 0x01
cnt_div: resb 0x01
result: resq 0x01
cnta: resb 0x01
cntd: resb 0x01
res: resb 0x01
cnt_mul: resb 0x01
;***************************************************
%macro print 2 ;macro for printing
mov rax,0x01
mov rdi,0x01
mov rsi,%1
mov rdx,%2
syscall
%endmacro
;***************************************************
%macro read 2 ;macro for reading
mov rax,0x00
mov rdi,0x00
mov rsi,%1
mov rdx,%2
syscall
%endmacro
;***************************************************
Section .text
Global _start
_start:
menu:
print msg1,len1
read chc,0x02
cmp byte[chc],0x31
je hex
cmp byte[chc],0x32
je bcd
cmp byte[chc],0x33
je exit
print msg6,len6
jmp menu
hex:
```

```
print msg2,len2
read temp,0x06
mov rsi,temp
mov byte[cnta],0x04 ;4 digit Input
call ascii_hex
print msg3,len3
call hextobcd
print msg,len
jmp menu
;****************************************************
bcd:
print msg4,len4
read temp,0x06
mov rsi,temp
mov byte[cnta],0x05 ;5 digit input
call ascii_hex
print msg5,len5
call bcdtohex
print msg,len
jmp menu
;****************************************************
exit:
mov rax,0x3C
mov rdi,0x00
syscall
;****************************************************
hextobcd:
xor eax,eax
xor ecx,ecx
mov ax,word[data]
mov byte[cnt_div],0x05
loop1:
xor edx,edx
mov bx,0x0A
div bx ;Result in ax Remainder in dx
ror ecx,0x04
or cx,dx
dec byte[cnt_div]
jnz loop1
rol ecx,0x10 ;Result in proper form
mov dword[result+0x04],ecx
mov byte[cntd],0x08
call disp
ret
;****************************************************
bcdtohex:
xor eax,eax
xor rcx,rcx
xor rbx,rbx
mov byte[cnt_mul],0x05
mov r8,[data]
ror r8,0x10 ;Get MSB at lowest nibble
back2:
mov bx,0x0A
mul bx
mov rcx,r8
and rcx,0xF ;Seperate req digit
add eax,ecx
rol r8,0x04
dec byte[cnt_mul]
jnz back2
mov dword[result+0x04],eax
mov byte[cntd],0x08
call disp
```

```
ret
;****************************************************
ascii_hex:
xor ebx,ebx
xor eax,eax
digit2:
mov bl,byte[rsi]
cmp bl,0x39
jbe digit1
sub bl,0x07
digit1:
sub bl,0x30
sal eax,0x04
add al,bl
inc rsi
dec byte[cnta]
jnz digit2
mov dword[data],eax
ret
disp:
xor rbx,rbx
back:
rol qword[result],0x04
mov bl,byte[result]
and bl,0FH
cmp bl,09H
jbe next
add bl,0x07
next:
add bl,0x30
mov byte[res],bl
print res,1
dec byte[cntd]
jnz back
ret
```

Outpu



```
rllab@fedora:/home/liveuser$ nasm -f elf64 prathamesh6.nasm
rllab@fedora:/home/liveuser$ ld -o prathamesh6 prathamesh6.o
rllab@fedora:/home/liveuser$ ./prathamesh6
1. HEX to BCD
2. BCD to HEX
3. Exit
Enter choice: 1

Enter 4-digit HEX number: 4875

Equivalent BCD: 00018549
1. HEX to BCD
2. BCD to HEX
3. Exit
Enter choice: 2

Enter 5-digit BCD number: 55465

Equivalent HEX: 0000D8A9
1. HEX to BCD
2. BCD to HEX
3. Exit
Enter choice: 3
rllab@fedora:/home/liveuser$
```