MIT LAB 6

SAHIL BONDRE: U18CO021

1. WAP to find Factorial of a given number using Call and Subroutine.

```
;<q1>

jmp start

;data
;code
start: nop
lxi SP, 4000H
mvi B, 4
nop
call factorial

hlt

; Factorial of B stored in D
; req: A
factorial: nop
mvi A, 1
cmp B
jnz next
mvi D, 1
ret
next: nop
dcr B
call factorial
inr B
mov C, D
call multiply
mov D, C
ret

; Multiply B & C and store in C
; req: A, B, C
multiply: nop
mvi A, 0
loop: nop
add B
dcr C
jnz loop
```

```
mov C, A
ret
```

factorial of B = 4 stored in D =18H = 24

2. WAP for Fibonacci Series using Call and Subroutine

```
;<q2>

jmp start

;data
;code
start: nop
lxi H, 4000H
mvi B, 07H
call fib

hlt

; calculates first [B] = n fibonacci numbers
; req: A, B, M, C
fib: nop
mvi C, 0 ; counter
loop: nop
mvi A, 00H
cmp C
jnz skip
mvi M, 00H
jmp next
skip: nop
mvi A, 01H
cmp C
```

```
jnz skip2
mvi M, 01H
jmp next
skip2: nop
dcx H
mov A, M
dcx H
add M
inx H
inx H
mov M, A
next: inx H
inr C
mov A, C
cmp B
jnz loop
ret
```

Address (Hex)	Address	Data
4000	16384	0
4001	16385	1
4002	16386	1
4003	16387	2
4004	16388	3
4005	16389	5
4006	16390	8
4007	16391	0
4008	16392	0

first 7 fibonacci numbers

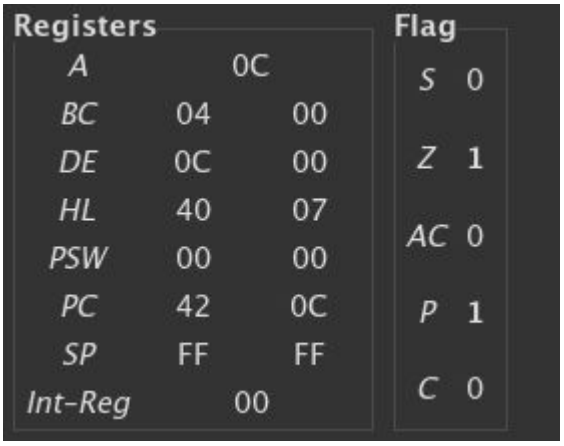
3. WAP to find Multiplication of Two 8-Bit Numbers using Call and Subroutine.

```
;<q3>

jmp start

;data
;code
start: nop
mvi B, 04H
mvi C, 03H
call multiply
hlt
```

```
; Multiply B & C and store in D
; req: A, B, C
multiply: nop
mvi A, 0
loop: nop
add B
dcr C
jnz loop
mov D, A
ret
```



04 x 03 = 0C (12) stored in A

4. Write Assembly language program to find the square/square root of a number .The number is stored at location 5000H, store the result at 5050H.

Square Root:

```
;<q4>

jmp start

;data


;code
start: nop
lxi H, 5000H
mov D, M
call sqrt
lxi H, 5050H
mov M, E
hlt
```

```
; Store sqrt of D in E
sqrt: nop
mvi H, 01H ; counter
loop1: nop
mov B, H
mov C, H
call multiply
mov A, C
cmp D
jnz end
mov E, B
ret
end: nop
inr H
mov A, D
cmp H
jnz loop1
ret


; Multiply B & C and store in C
; req: A, B, C
multiply: nop
mvi A, 0
loop: nop
add B
dcr C
jnz loop
mov C, A
ret
```

Address (Hex)	Address	Data
5000	20480	49
5001	20481	0
5002	20482	0
5003	20483	0

5000H has 49

Address (Hex)	Address	Data
504E	20558	0
504F	20559	0
5050	20560	7
5051	20561	0
5052	20562	0

5050H has sqrt(49) = 7

Square:

```
;<q4b>

jmp start

;data


;code
start: nop
lxi H, 5000H
mov D, M
call sqr
lxi H, 5050H
mov M, C
hlt

; Store sqr of D in C
sqr: nop
mov B, D
mov C, D
call multiply
ret

; Multiply B & C and store in C
; req: A, B, C
multiply: nop
mvi A, 0
loop: nop
add B
dcr C
```

```
jnz loop
mov C, A
ret
```

Registers			Flag	
A	31		S	0
BC	07	31	Z	1
DE	07	07	AC	0
HL	50	50	P	1
PSW	00	00	C	0
PC	42	10		
SP	FF	FF		
Int-Reg	00			

Address (Hex)	Address	Data
5000	20480	7
5001	20481	0
5002	20482	0
5003	20483	0

5000H has 7

504F	20559	0
5050	20560	49
5051	20561	0
5052	20562	0

5050H has sqr(7) = 49