MIT LAB 7

U18CO021: SAHIL BONDRE

1. Write a program to convert a given number of binary data bytes into their BCD equivalents, and store them as unpacked BCDs in the output buffer. The number of data bytes is specified in register D in the main program. The converted numbers should be stored in groups of three consecutive memory locations. If the number is not large enough to occupy all three locations, Zeros should be loaded in those locations.

```
;<q1>

jmp start

;code
start: nop
mvi D, 03H
lxi B, 2000H; start location
lxi H, 3000H; end location
first: ldax B
call pwr
inx h
inx b
dcr d
jnz first
hlt


pwr: nop
mvi e, 64H
call bcd
mvi e, 0AH
call bcd
mov m, a
ret


bcd: nop
mvi m, 0FFH
```

```
next: inr m
sub e
jnc next
add e
inx h
ret
```

Address (Hex)	Address	Data
2000	8192	255
2001	8193	198
2002	8194	10
2003	8195	0

Decimal

Address (Hex)	Address	Data
3000	12288	2
3001	12289	5
3002	12290	5
3003	12291	1
3004	12292	9
3005	12293	8
3006	12294	0
3007	12295	1
3008	12296	0
3009	12297	0
300A	12298	0

BCD

2. A set of ten BCD readings is stored in the Input Buffer. Convert the numbers into binary and add the numbers. Store the sum in the Output Buffer, the sum can be larger than FFH.

```
;<q2>

jmp start

;code
start: nop
mvi E, 0AH
lxi SP, 7000H
lxi H, 4000H
lxi B, 5000H
first: mov A, M
call bcd
stax B
inx B
inx H
dcr E
jnz first
xra A
mvi E, 0AH
lxi B, 5000H
lxi H, 0000H
loop: ldax B
push D
mov E, A
dad D
inx B
pop D
dcr E
jnz loop
shld 6000H
hlt


bcd: push B
push D
mov B, A
ani 0FH
mov C,A
mov A, B
ani 0F0H
jz bcd1
```

```
rrc
rrc
rrc
rrc
mov D, A
xra A
mvi E, 0AH
sum: add E
dcr D
jnz sum
bcd1: add C
pop D
pop B
ret
```

Address (Hex)	Address	Data
4000	16384	12
4001	16385	59
4002	16386	11
4003	16387	45
4004	16388	30
4005	16389	11
4006	16390	1
4007	16391	2
4008	16392	66
4009	16393	3

BCD Values

Address (Hex)	Address	Data
5000	20480	12
5001	20481	41
5002	20482	11
5003	20483	33
5004	20484	24
5005	20485	11
5006	20486	1
5007	20487	2
5008	20488	42
5009	20489	3
500A	20490	0

Binary Conversion

Address (Hex)	Address	Data
6000	24576	180
6001	24577	0
6002	24578	0
6003	24579	0
6004	24580	0

Sum

3. A set of ASCII Hex digits is stored in the Input Buffer memory. Write a program to convert these numbers into binary. Add these numbers in binary, and store the result in the Output-Buffer memory.

```
;<q3>

jmp start

;code
start: nop
mvi D,03H
lxi H,4000H
lxi B,5000H
first: nop
mov A, M
call asctobin
stax B
dcr D
inx H
inx B
jnz first
lxi H, 5000H
mvi D, 03H
mvi A, 00H
up: mov E, M
add E
dcr D
inx H
jnz up
sta 6000H
hlt

asctobin: sui 30H
cpi 0AH
rc
sui 07H
```

```
ret
```

Address (Hex)	Address	Data
4000	16384	50
4001	16385	55
4002	16386	53
4003	16387	0
4004	16388	0
4005	16389	0

ASCII

Start 5000h

Address (Hex)	Address	Data
5000	20480	2
5001	20481	7
5002	20482	5
5003	20483	0
5004	20484	0
5005	20485	0
5006	20486	0

BIN

Start 6000h

Address (Hex)	Address	Data
6000	24576	14
6001	24577	0
6002	24578	0
6003	24579	0

SUM