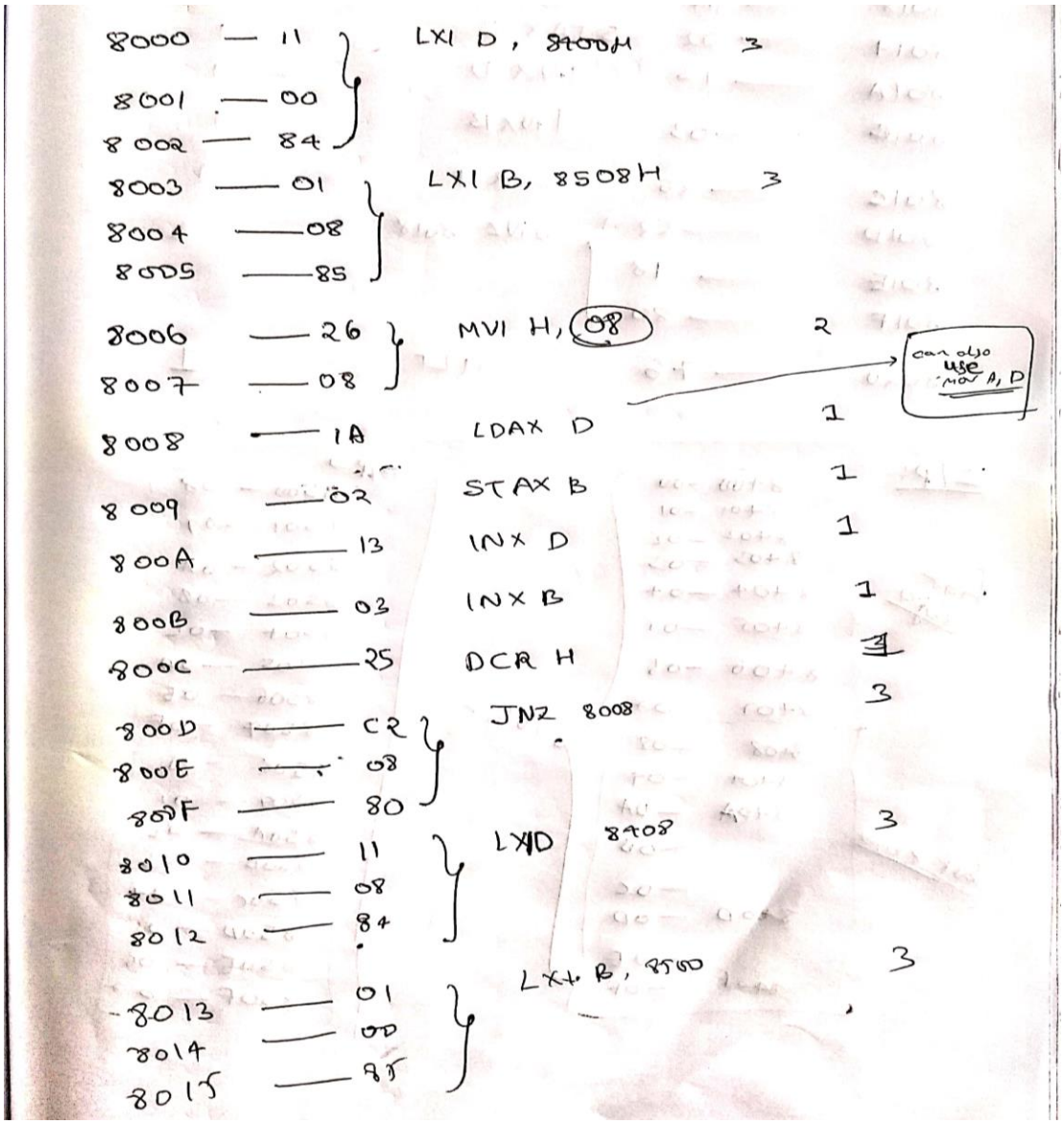
8085
PROGRAMS

Experiment - 1

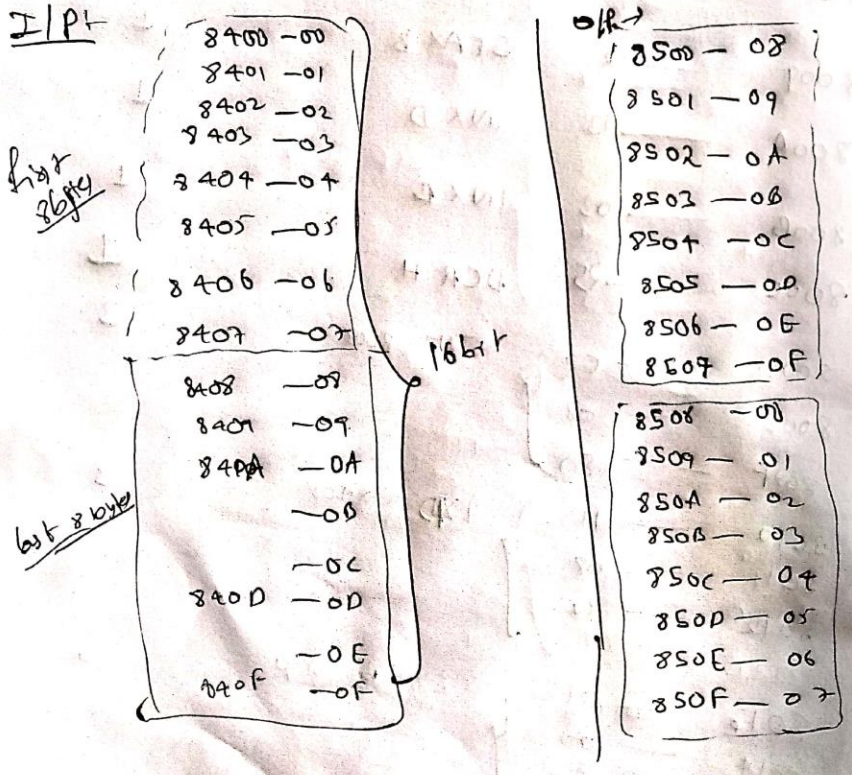
REARRANGING BYTES:

16 bytes are residing in locations starting from 0C20H / 2400H. Write a program to transfer these bytes in locations starting from 0C40H / 2500H in such a way that first 8 bytes in the first block should appear at last 8 locations of the second block and the last 8 bytes in the first block at first 8 locations of the second block .

CODE:



8016	26	MVI H (08)	2
8017	08		
8018	1A	LDAX D	1
8019	02	STAX B	1
801A	13	INX D	1
801B	03	INX B	1
801C	25	DCR H	1
801D	C2	JNZ 8018	3
801E	18		
801F	80		
8020	76	HLT	1

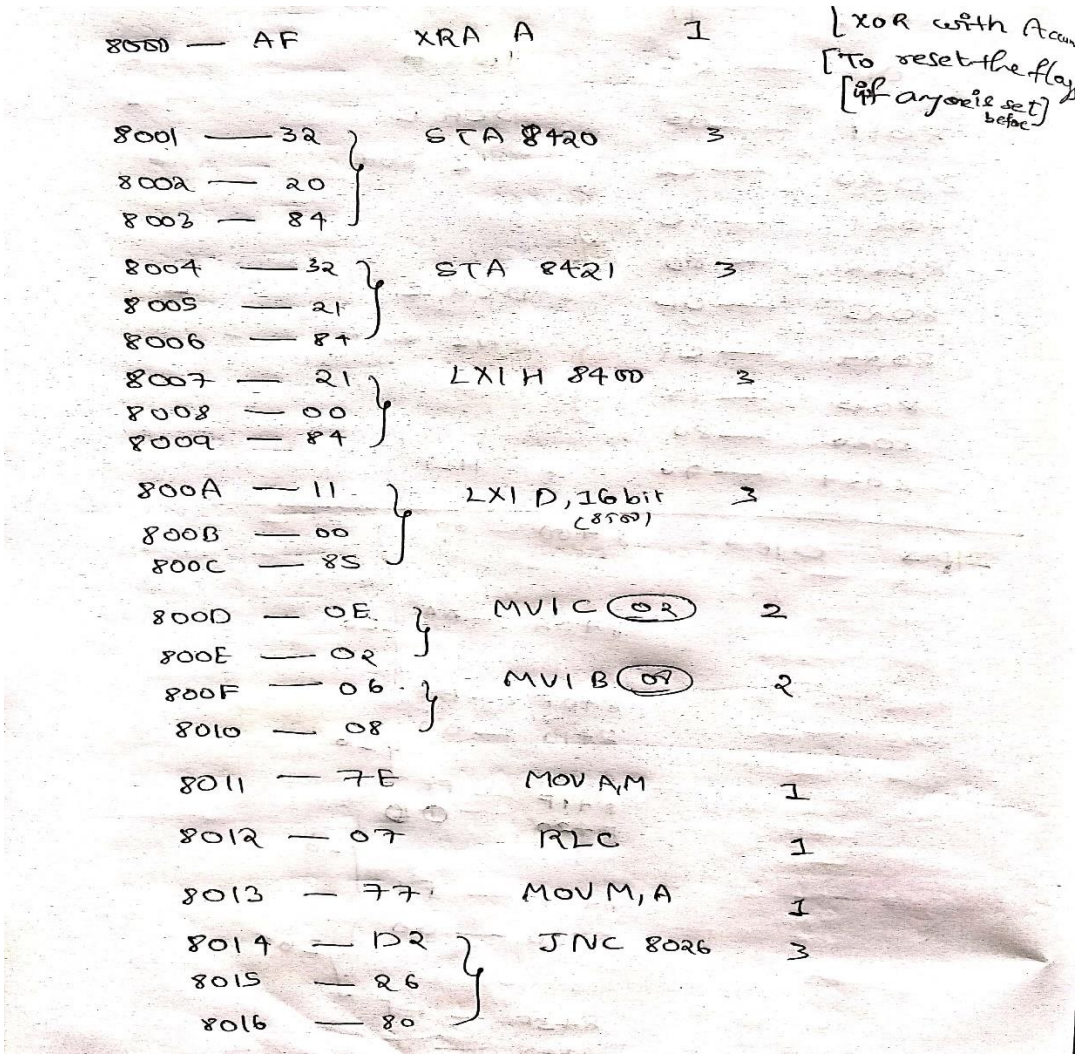


Experiment - 2

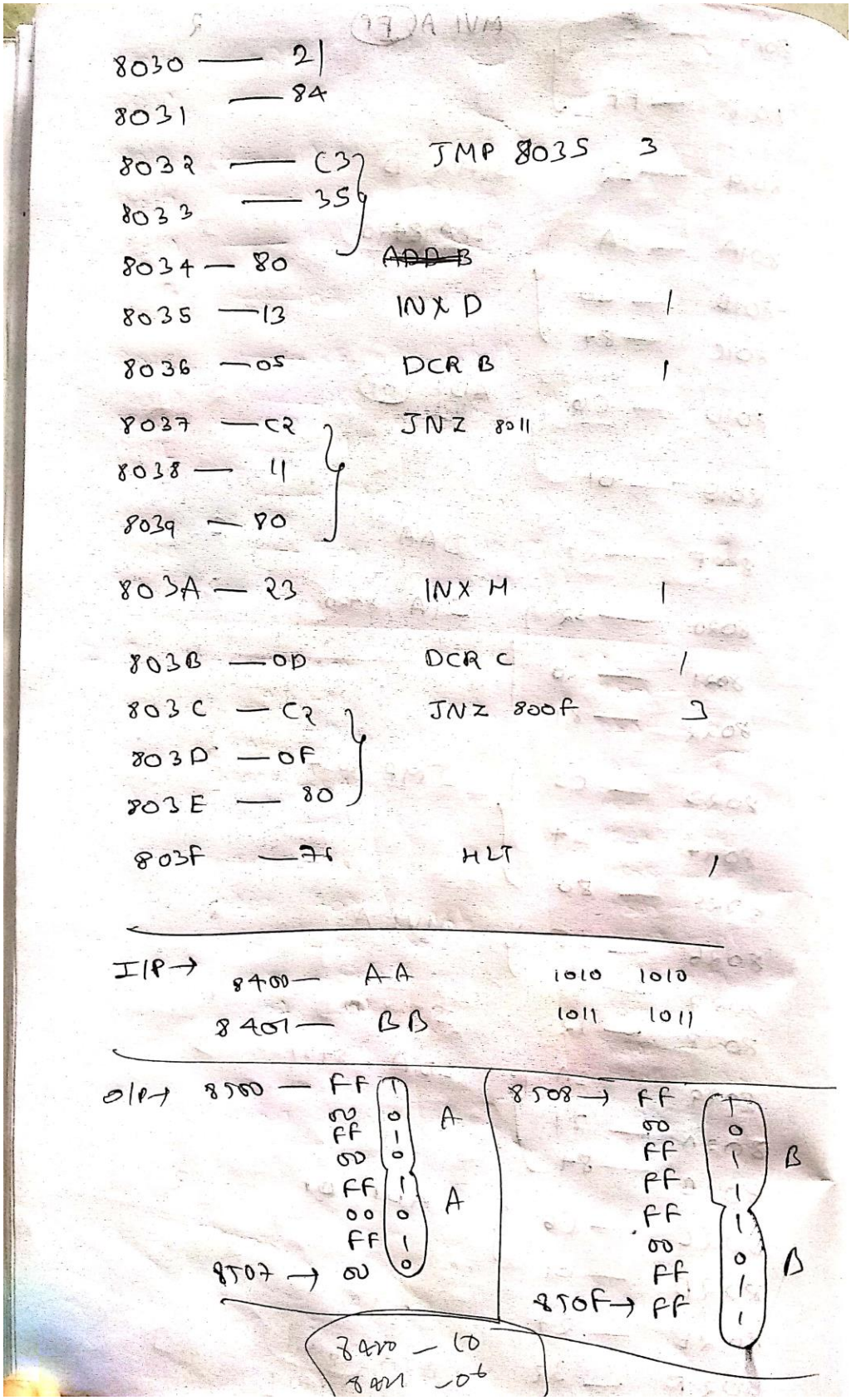
CHECKING BITS OF A WORD:

A word is residing in location 0C40H / 2400H. Write a program to check each bit of the word starting from Ms bit and fill 16 locations starting from 0C20H / 2500H, with either 00H or FFH depending on the bit, FFH if bit is '1' and 00H if the bit is '0'. Also count the no. of 1's and 0's (count in BCD) in the word and store them respectively at 0C30H / 2420H and 0C31H / 2421H

CODE:



8017 — 3E	MVI A FF	2
8018 — FF		
8019 — 12	STAX D	1
801A — 3A	LDA 8420	3
801B — 20		
801C — 84		
801D — C6	ADI 8bit 01	2
801E — 01		
801F — 27	DAA	1
8020 — 32	STA 8420	3
8021 — 20		
8022 — 84		
8023 — C3	JMP 8034	3
8024 — 34		
8025 — 80		
8026 — 3E	MVI A	2
8027 — 00		
8028 — 12	STAX D	1
8029 — 3A	LDA 8421	3
802A — 21		
802B — 84		
802C — C6	ADI 01	2
802D — 01		
802E — 27	DAA	
802F — 32	STA 16 bit 8421	3

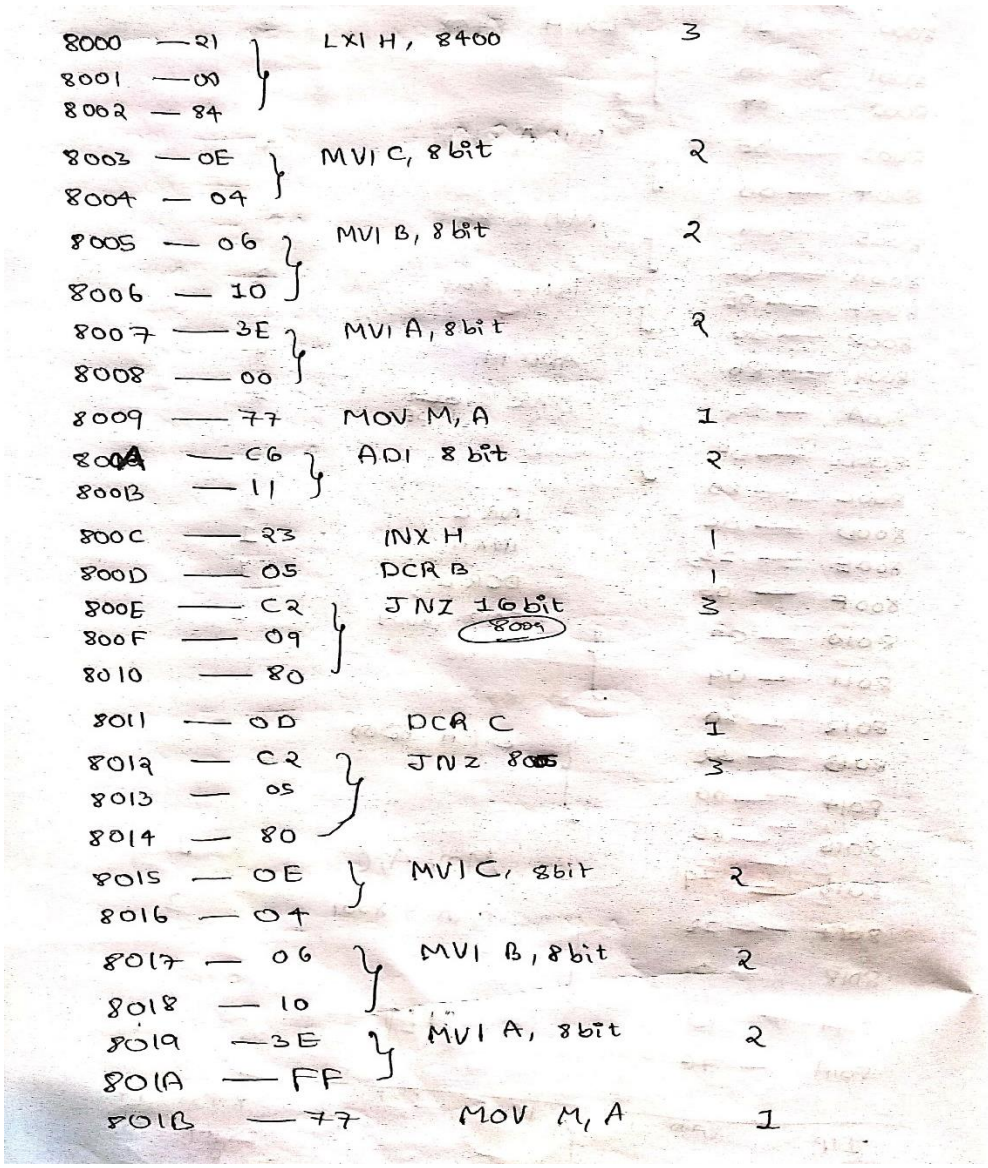

8030 — 21
8031 — 84
8032 — C3
8033 — 35
JMP 8035 3
8034 — 80
ADD B
8035 — 13
INX D 1
8036 — 05
DCR B 1
8037 — C2
8038 — 11
8039 — 80
JNZ 8011
803A — 23
INX H 1
803B — 0D
DCR C 1
803C — C2
803D — 0F
803E — 80
JNZ 800f 3
803f — 76
HLT 1
I/P →
8400 — AA
8401 — BB
1010 1010
1011 1011
O/P →
8500 — FF
8507 →
8508 →
8509 →
8500 — 10
8501 — 06

Experiment-3

FILLING UP 128 LOCATIONS:

Write a program to fill up 128 locations starting from 0C00H / 2400H with bytes in the following pattern: first 64 locations to be filled up as 00H, 11H, 22H, ------, FFH, 00H, 11H, 22H, --------- and so on and the last 64 locations to be filled up as FFH, EEH, DDH, ---------, 00H, FFH, EEH, DDH and so on….

CODE :



801C — D6	SUI Data (D1)		2
801D — 11			
801E — 23	INX H		1
801F — 05	DCR B		1
8020 — C2	JNZ 801B		3
8021 — 1B			
8022 — 80			
8023 — 0D	DCR C		1
8024 — C2	JNZ 8017		3
8025 — 17			
8026 — 80			
8027 — 76	HLT		

~~I/P~~ O/P →

8400 — 00
11
22
:
840A — AA
:
840F — FF } 16

8410 — 00
:
841F — FF } 16

8420 — 00
:
842F — FF } 16

8450 — 00
:
843F — FF } 16

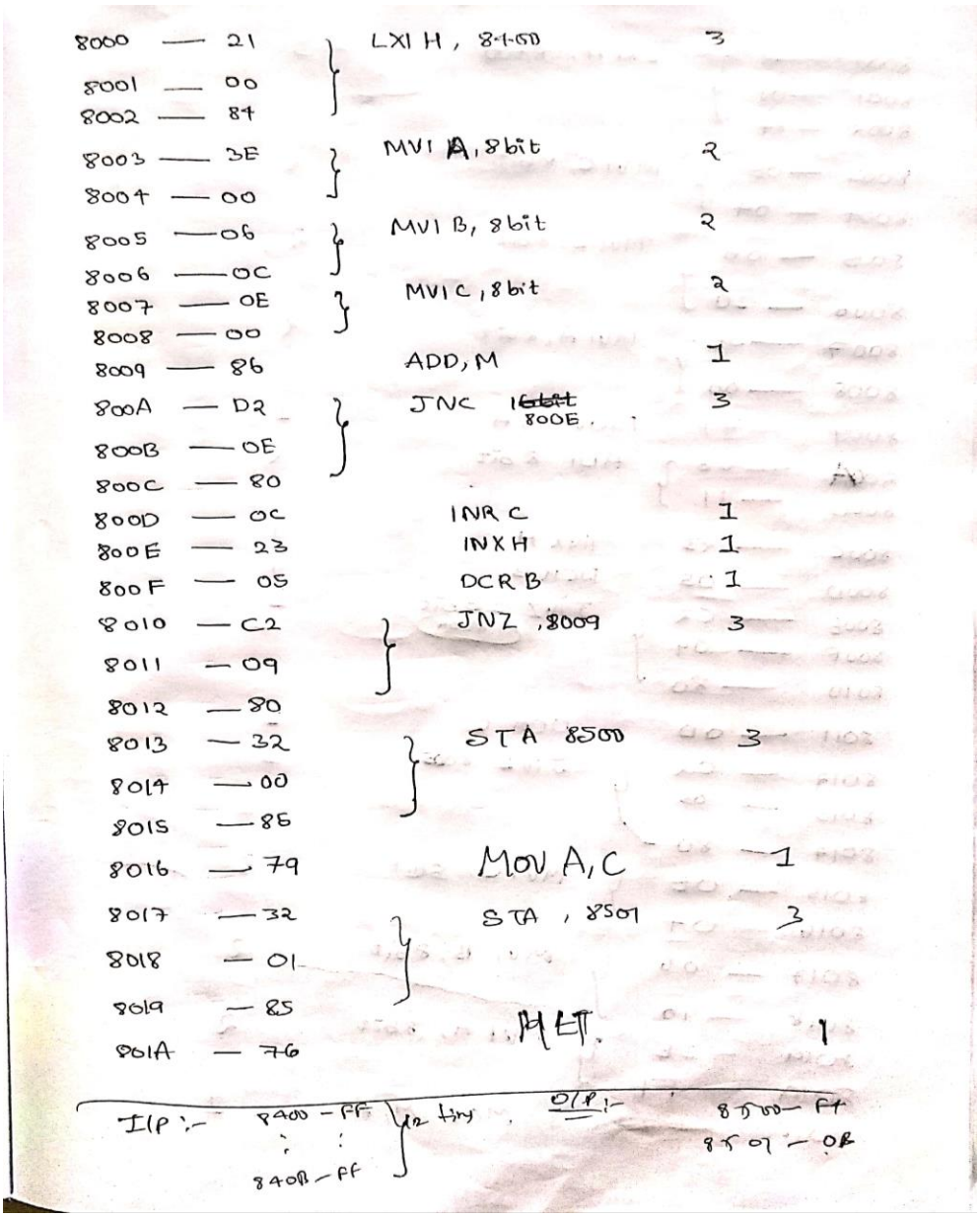
8440 — FA
:
847F — 00 } 64

EXPERIMENT-4

ADDITION OF 12 BYTES:

Write a program to add 12 bytes residing in locations starting from 0C20H / 2400H, and store the sum in location 0C30H / 2500H (result space 2 bytes)

CODE:

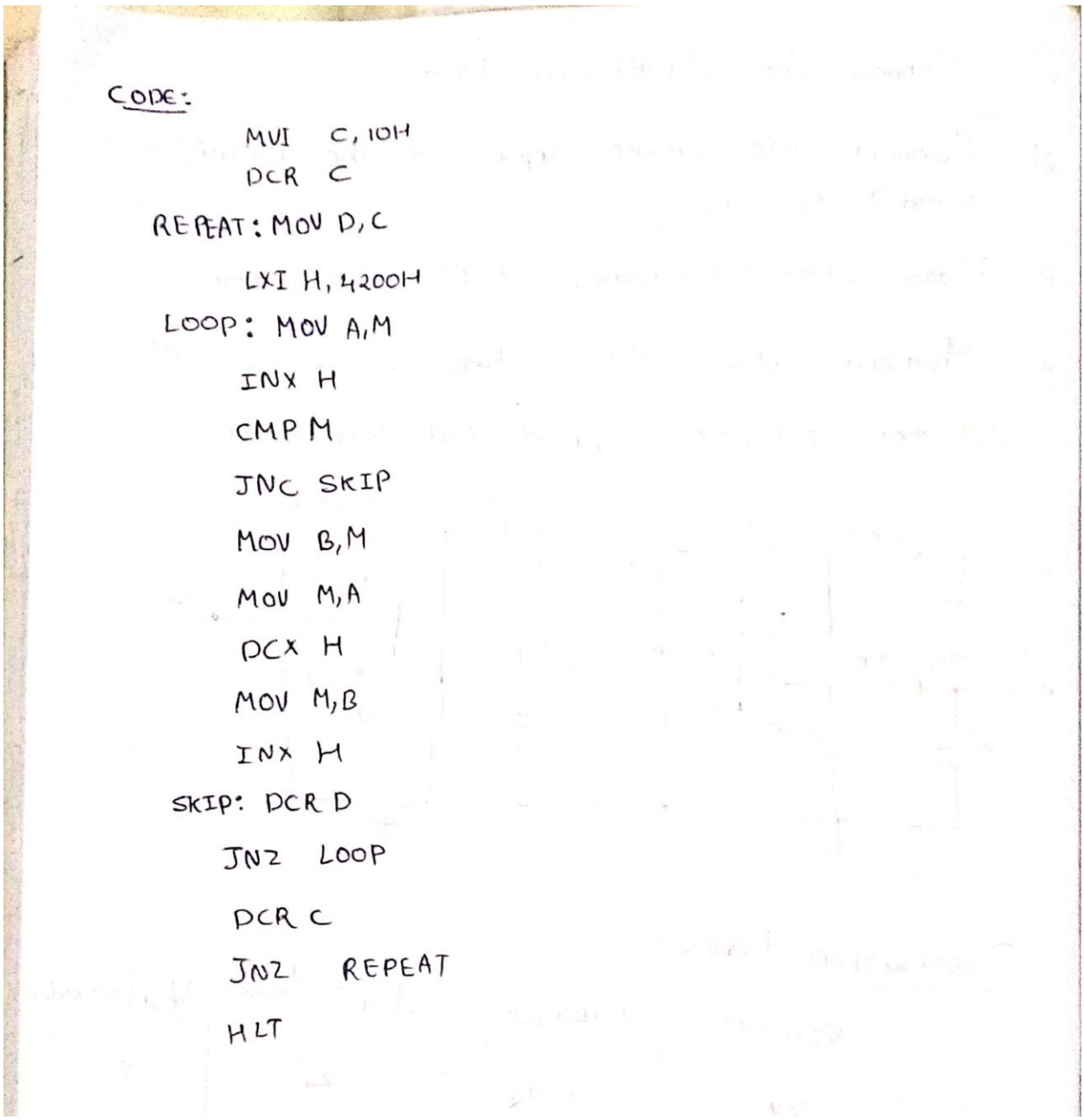


EXPERIMENT-5

SORTING IN DESCENDING ORDER

Write a program to sort 16 bytes residing in locations starting from 0C20H / 2400H, in descending order in the same location .

CODE:



```
CODE:
        MVI   C, 10H
        DCR   C
REPEAT: MOV D, C
        LXI H, 4200H
LOOP:   MOV A, M
        INX H
        CMP M
        JNC SKIP
        MOV B, M
        MOV M, A
        DCX H
        MOV M, B
        INX H
SKIP:   DCR D
        JNZ LOOP
        DCR C
        JNZ REPEAT
        HLT
```

INPUT :

Memory View 0x 4200

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
420	00	00	0F	11	22	22	33	34	34	56	76	88	99	AB	FF	FF
421	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
422	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
423	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
424	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
425	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
426	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
427	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
428	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
429	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 3300

OUTPUT :

Memory View 0x 4200

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
420	FF	FF	AB	99	88	76	56	34	34	33	22	22	11	0F	00	00
421	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
422	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
423	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
424	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
425	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
426	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
427	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
428	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
429	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

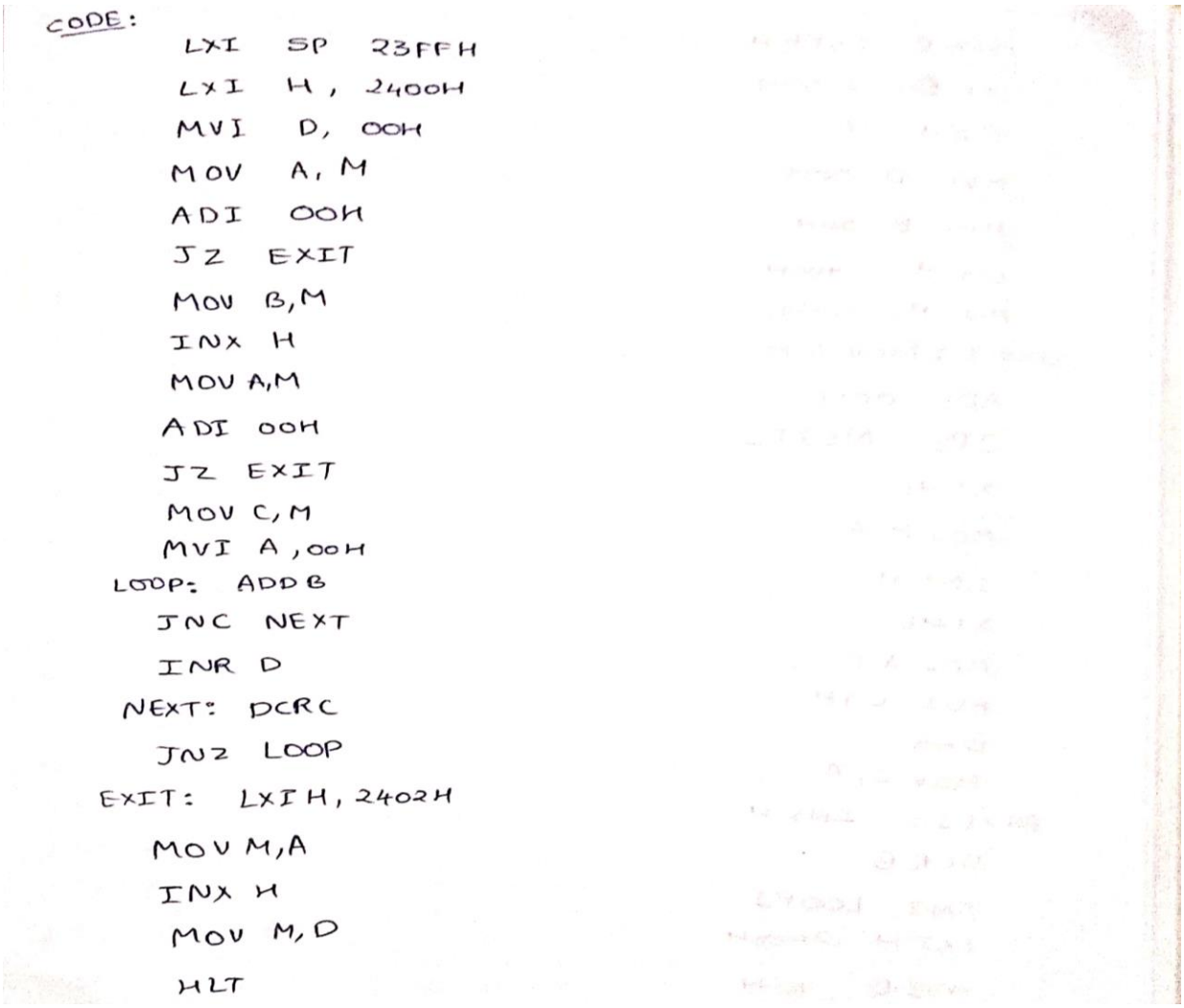
Start Address at: 0x 3300

Experiment -6

MULTIPLICATION OF SINGLE BYTE BY SINGLE BYTE:

Write a program to multiply a single byte residing in location 0C20H / 2400H by another byte residing in location 0C21H / 2401H and store the product in 0C22H / 2402H (Product space 2 bytes)

CODE:



```
CODE:
        LXI   SP  23FFH
        LXI   H , 2400H
        MVI   D, 00H
        MOV   A, M
        ADI   00H
        JZ    EXIT
        MOV   B,M
        INX   H
        MOV A,M
        ADI   00H
        JZ    EXIT
        MOV C,M
        MVI   A ,00H
LOOP:   ADD B
        JNC   NEXT
        INR   D
NEXT:   DCR C
        JNZ   LOOP
EXIT:   LXI H, 2402H
        MOV M,A
        INX H
        MOV M, D
        HLT
```

INPUT/OUTPUT:

Memory View

0x 2400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	FF	FF	01	FE	00	00	00	00	00	00	00	00	00	00	00	00
241	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
242	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

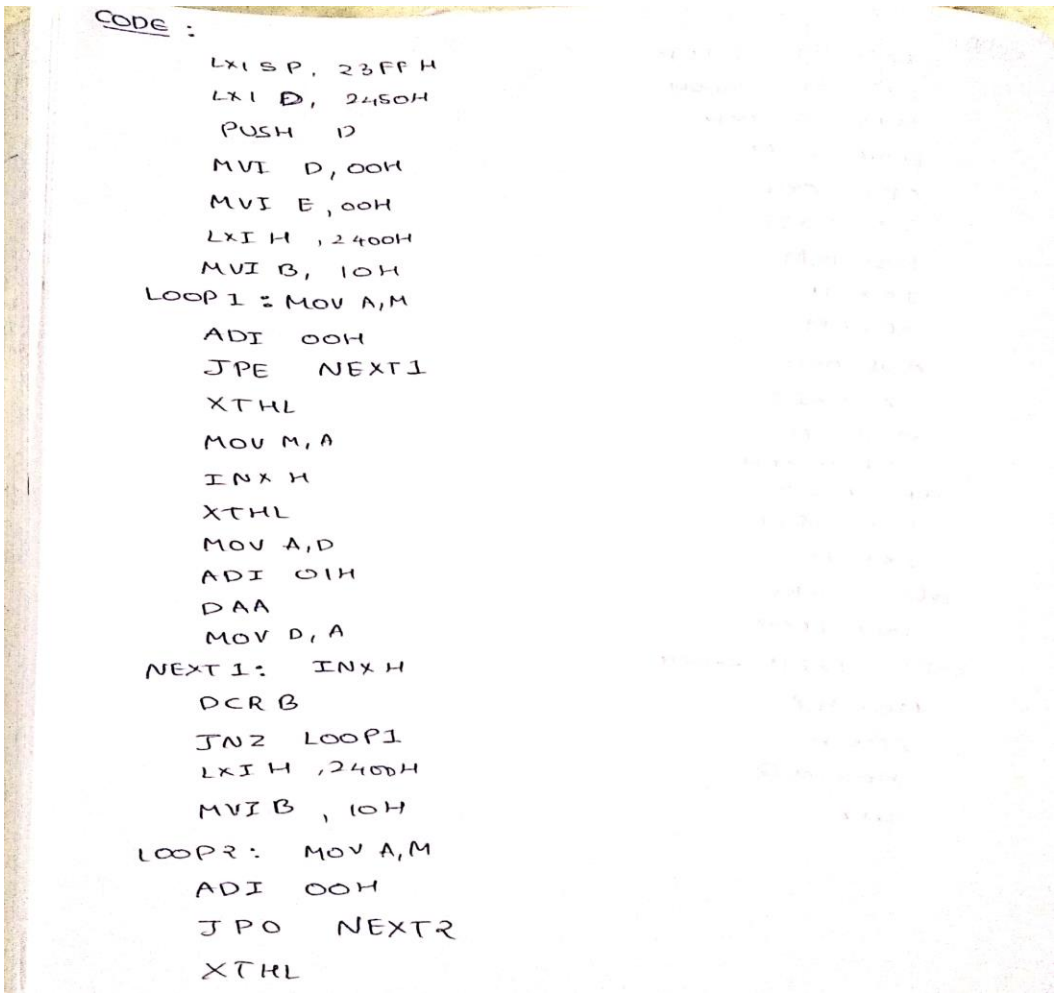
Start Address at: 0x 2200

Experiment - 7

SORTING EVEN AND ODD PARITY BYTES:

Sixteen bytes reside in locations starting from 0C20H / 2400H. Write a program to sort them in the same locations according to parity, odd parity bytes first and then even parity bytes next. Keep the count (in BCD) of odd parity and even parity bytes at 0C30H / 2420H and 0C31H / 2421H respectively

CODE:



```
CODE :
        LXI SP, 23FFH
        LXI D, 2450H
        PUSH D
        MVI D, 00H
        MVI E, 00H
        LXI H, 2400H
        MVI B, 10H
LOOP1 : MOV A,M
        ADI 00H
        JPE NEXT1
        XTHL
        MOV M, A
        INX H
        XTHL
        MOV A,D
        ADI 01H
        DAA
        MOV D, A
NEXT1 : INX H
        DCR B
        JNZ LOOP1
        LXI H, 2400H
        MVI B, 10H
LOOP2 : MOV A,M
        ADI 00H
        JPO NEXT2
        XTHL
```

```
        MOV  M,A
        INX H
        XTHL
        MOV A,E
        ADI  01H
        DAA
        MOV E,A
NEXT2:  INX H
        DCR B
        JNZ LOOP2
        LXI H ,2420H
        MOV M,D
        INX H
        MOV M,E
        LXI H ,2400H
        LXI D, 2450H
        MVI B, 10H
LOOP3:  LDAX D
        MOV M, A
        INX D
        INX H
        DCR B
        JNZ  LOOP3
        HLT
```

INPUT/OUTPUT :

Memory View

0x 2400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	01	02	04	07	08	0B	0D	0E	03	05	06	09	0A	0C	0F	00
241	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
242	2C	38	00	00	00	00	00	00	00	00	00	00	00	00	00	00
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	01	02	04	07	08	0B	0D	0E	03	05	06	09	0A	0C	0F	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

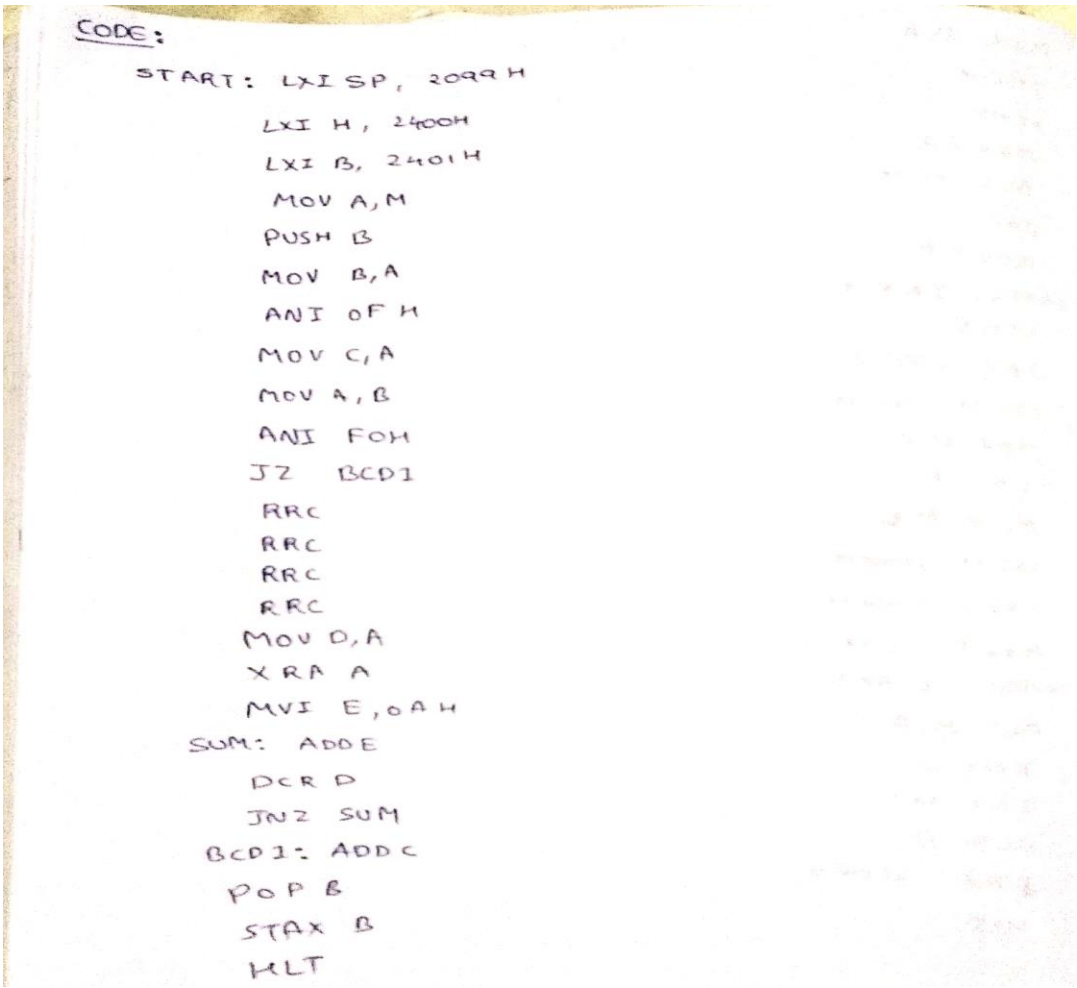
Start Address at: 0x 2200

Experiment - 8

BCD TO BINARY CONVERSION:

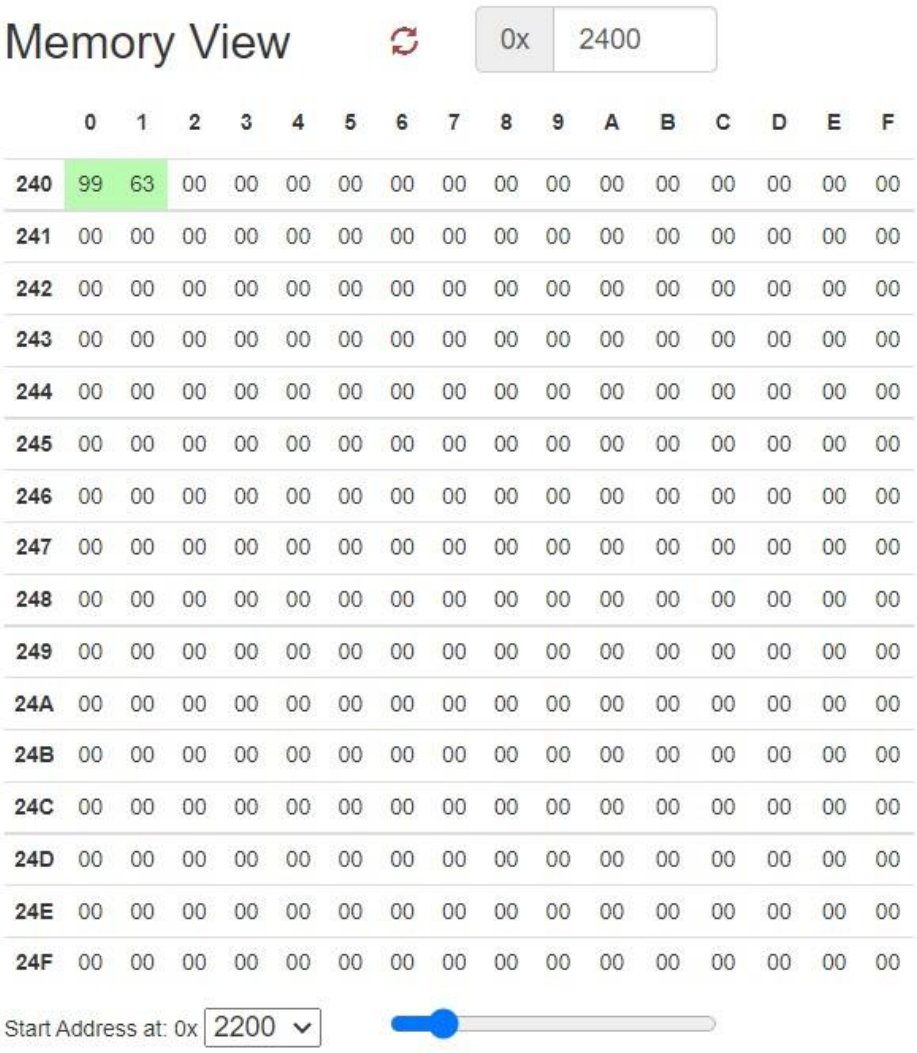
Write a program to convert a BCD number residing at 0C20H / 2400H to its corresponding binary number and store it at 0C21H / 2401H

CODE:



```
CODE:
START: LXI SP, 2099H
       LXI H, 2400H
       LXI B, 2401H
       MOV A,M
       PUSH B
       MOV B,A
       ANI 0FH
       MOV C,A
       MOV A,B
       ANI F0H
       JZ  BCD1
       RRC
       RRC
       RRC
       RRC
       MOV D,A
       XRA A
       MVI E,0AH
SUM:   ADDE
       DCR D
       JNZ SUM
BCD1:  ADDC
       POP B
       STAX B
       HLT
```

INPUT/OUTPUT:



Experiment - 9

NUMBER OF BITS IN BYTES:

A block of 16 bytes resides in locations starting from 0C20H / 2400H. Write a program to form a second block of size 16 with the no. of 1's in the corresponding byte of the first block, at locations starting from 0C30H / 2420H. (For e.g. the first entry in the second block will be the no. of 1's in the first entry of the first block.)

CODE:

```
CODE:   LXI  SP, 23FFH
        LXI  H   2400H
        LXI  D,  2420H
        PUSH D
        MVI  B, 10H
LOOP1:  MVI  D, 00H
        MVI  C, 08H
        MOV  A, M
LOOP2:  MOV  E, A
        ANI  01H
        JZ   NEXT
        INR  D
NEXT:   MOV  A, E
        RAR
        DCR  C
        JNZ  LOOP2
        XTHL
        MOV  M, D
        INX  H
        XTHL
        INX  H
        DCR  B
        JNZ  LOOP1
        HLT
```

INPUT/OUTPUT :

Memory View

0x 2400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
241	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
242	00	01	01	02	01	02	02	03	01	02	02	03	02	03	03	04
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

Experiment -10

NUMBER OF TIMES FFH OCCURS AS PAIR:

A block starts at 0C20H / 2400H and ends at 0C2FH / 240FH. Write a program to count the number of times FFH occur as memory pairs and keep the count (BCD) in location 0C30H / 2410H. If four consecutive locations are filled up with FFH the count shall be 2 and so on

CODE:

CODE:

```
        LXI H, 2400H
        MVI B, 0FH
        MVI C, 00H
LOOP:   MOV A,M
        CPI FFH
        JNZ NEXT
        INX H
        MOV A,M
        CPI FFH
        JNZ NEXT
        MOV A,C
        ADI 01H
        DAA
        MOV C,A
        DCR B
NEXT:   INX H
        DCR B
        JNZ LOOP
        LXI H, 2420H
        MOV M,C
        HLT
```

INPUT/OUTPUT :

Memory View

0x 2400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	FF	00	FF	FF	23	12	11	FF	FF	56	78	FF	FF	12	34	69
241	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
242	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

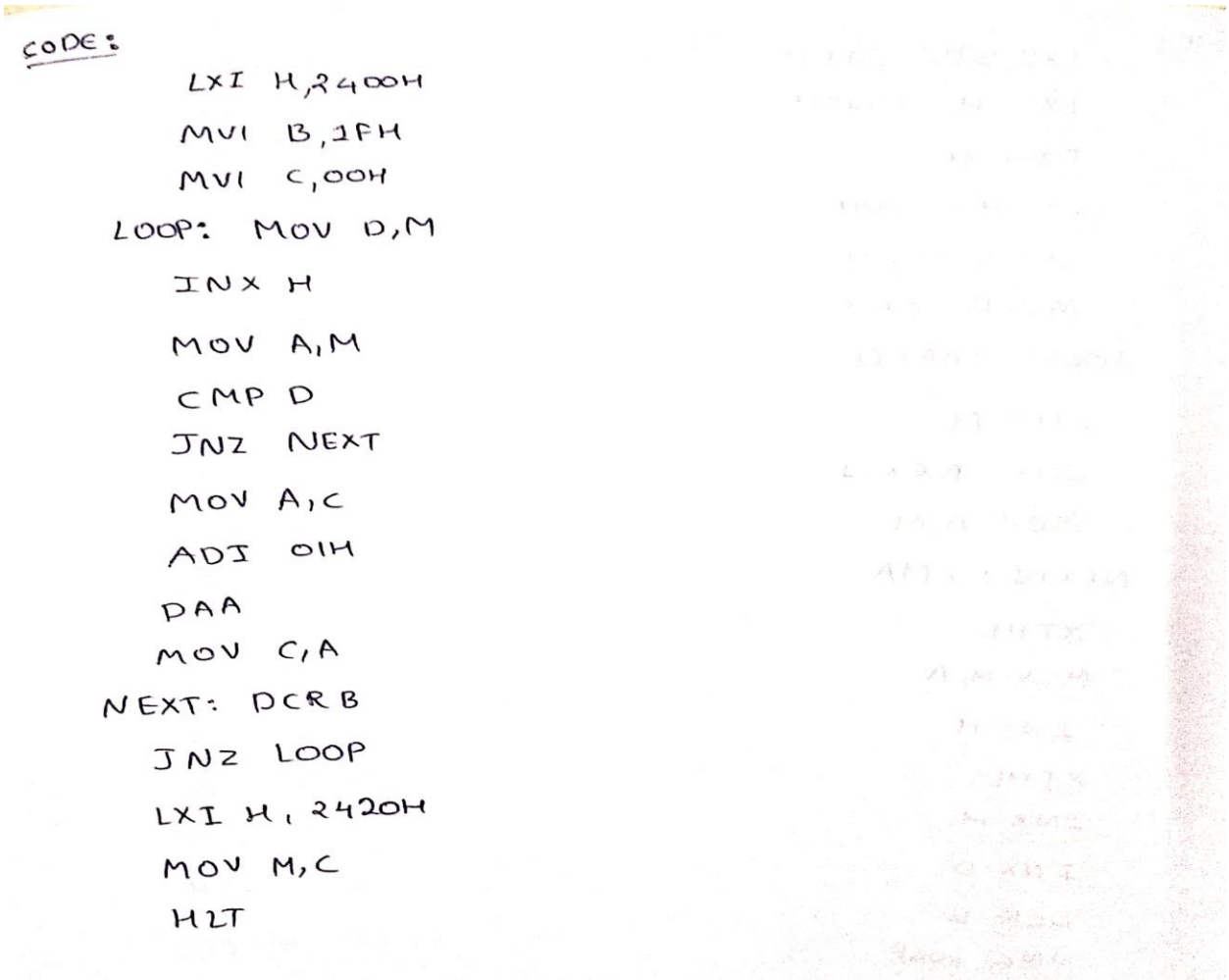
Start Address at: 0x 2200

EXPERIMENT-11

CONSECUTIVE MEMORY LOCATIONS WITH IDENTICAL DATA:

A block starts at 0C20H / 2400H and ends at 0C3FH / 241FH. Write a program to determine the number of consecutive locations with identical data and store the count (in BCD) in location 0C40H /2420H. If three consecutive locations have identical data, the count shall be 2 and so on

CODE:



```
CODE:
        LXI H,2400H
        MVI B,1FH
        MVI C,00H
LOOP:   MOV D,M
        INX H
        MOV A,M
        CMP D
        JNZ NEXT
        MOV A,C
        ADI 01H
        DAA
        MOV C,A
NEXT:   DCR B
        JNZ LOOP
        LXI H, 2420H
        MOV M,C
        HLT
```

INPUT/OUTPUT:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	FF	12	12	FF	99	FF	23	23	00	00	99	99	12	0A	AB	CD
241	AA	AA	34	69	90	45	22	22	11	23	56	65	34	34	12	12
242	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

EXPERIMENT-12

FORMATION OF A THIRD BLOCK:

Two blocks of 10 bytes each are residing in locations 0C20H / 2400H and 0C30H / 2410H respectively, Write a program to form a third block of 10 bytes at locations starting from 0C40H / 2420H, such that each entry in that block is the one's complement of the larger of the corresponding entries of the first two blocks (the n^{th} entry in the third block shall be the one's complement of the larger of the n^{th} entry of the first block and the n^{th} entry of the second block)

CODE:

CODE:

```
        LXI SP, 23FFH
        LXI H, 2420H
        PUSH H
        LXI H, 2400H
        LXI D, 2410H
        MVI B, 0AH
LOOP:   LDAX D
        CMP M
        JNC NEXT1
        MOV A,M
NEXT1 : CMA
        XTHL
        MOV M, A
        INX H
        XTHL
        INX H
        INX D
        DCR B
        JNZ LOOP
        HLT
```

INPUT/OUTPUT:

Memory View

0x 2400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	FF	12	12	FF	99	FF	23	23	45	54	00	00	00	00	00	00
241	AA	AA	34	69	90	45	22	22	11	23	00	00	00	00	00	00
242	00	55	CB	00	66	00	DC	DC	BA	AB	00	00	00	00	00	00
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

-

Experiment -13

INSERTION TO A LIST:

Write a program to insert the content at the location 0C20H / 2400H to a list which begins at 0C22H / 2402H, if not already present in the list and only if byte at location 0C20H / 2400H is divisible by two. The number of bytes in the list is given at location 0C21H / 2401H (in BCD). If the insertion is valid, insert the byte at the end of the list and modify the count accordingly

CODE:

```
CODE:
        LXI H,2401H
        MOV A,M
        CALL BCD
        CPI 00H
        JZ EXIT
        MOV B,A
        LDA 2400H
        RRC
        JC EXIT
        RLC
LOOP:   INX H
        CMP M
        JZ EXIT
        DCR B
        JNZ LOOP
        INX H
        MOV M,A
        LXI H,2401H
        MOV A,M
        ADI 01H
        DAA
        MOV M,A
EXIT: HLT
```

-

Experiment -13

INSERTION TO A LIST:

Write a program to insert the content at the location 0C20H / 2400H to a list which begins at 0C22H / 2402H, if not already present in the list and only if byte at location 0C20H / 2400H is divisible by two. The number of bytes in the list is given at location 0C21H / 2401H (in BCD). If the insertion is valid, insert the byte at the end of the list and modify the count accordingly

CODE:

```
CODE:
        LXI H,2401H
        MOV A,M
        CALL BCD
        CPI 00H
        JZ EXIT
        MOV B,A
        LDA 2400H
        RRC
        JC EXIT
        RLC
LOOP:   INX H
        CMP M
        JZ EXIT
        DCR B
        JNZ LOOP
        INX H
        MOV M,A
        LXI H,2401H
        MOV A,M
        ADI 01H
        DAA
        MOV M,A
EXIT: HLT
```

```
BCD:  ANI  0FH
      MOV  B,A
      MOV  A,M
      RRC
      RRC
      RRC
      RRC
      ANI  0FH
      MOV  C,A
      MOV  A,B
      INR  C
MYLOOP: DCR C
      JZ EXIT1
      ADI  0AH
      JMP  MYLOOP
EXIT1 : RET
```

INPUT:

Memory View 0x 2400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	10	09	01	02	03	FF	23	56	77	AB	CC	00	00	00	00	00
241	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
242	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

OUTPUT:

Memory View 0x 2400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
240	10	10	01	02	03	FF	23	56	77	AB	CC	10	00	00	00	00
241	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
242	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
243	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
244	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
245	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
246	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
247	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
249	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
24F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

8086
PROGRAMS

Experiment – 14

CHECKING BITS OF A WORD:

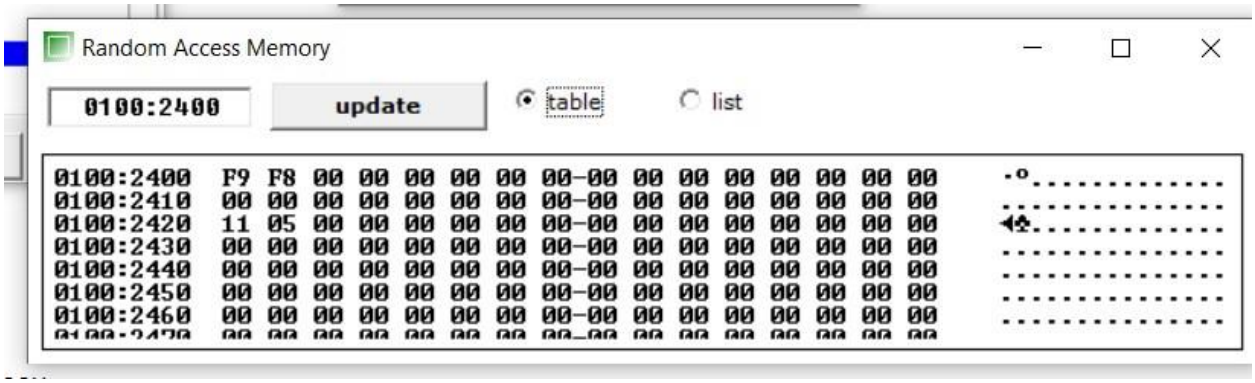
A word is residing in location 0C40H / 2400H .Write a program to check each bit of the word starting from Ms bit and fill 16locations starting from 0C20H / 2500H, with either 00H or FFH depending on the bit, FFH if bit is '1' and 00H if the is '0'. Also count the no. of 1's and 0's(count in BCD) in the word and store them respectively at 0C30H / 2420H and 0C31H / 2421H.

SOFTWARE USED: EMU8086 Simulator

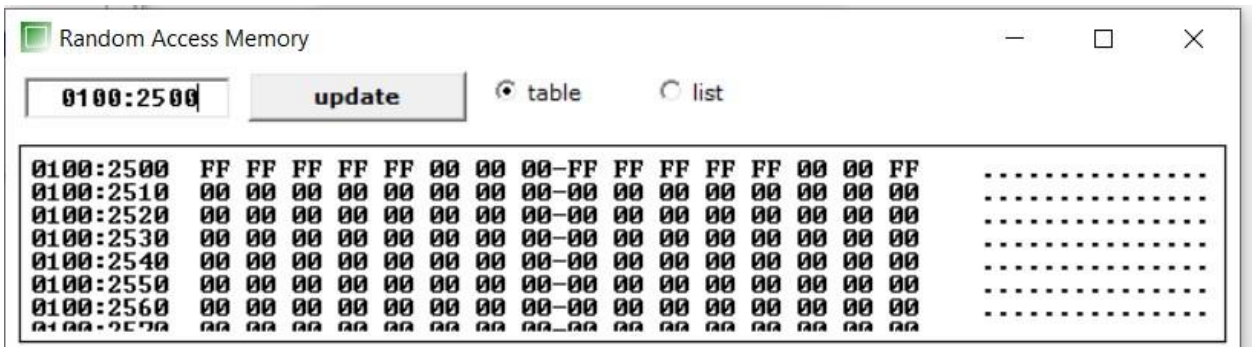
CODE:

```
CODE:
        MOV SI , 2500H
        MOV BL, [2400H]
        MOV BH, [2401H]
        MOV CL , 10H
        MOV AL, 00H
        MOV AH, 00H
BACK:   ROL  BX , 01H
        JC  NEXT
        MOV [SI] ,00H
        INC SI
        ADD  AL, 01H
        DAA
        DEC CL
        JNZ BACK
        JMP FINAL
NEXT:   MOV [SI] ,00FFH
        INC SI
        XCHG AL , AH
        ADD AL, 01H
        DAA
        XCHG AL,AH
        DEC CL
        JNZ BACK
FINAL:  MOV  [2420H], AH
        MOV [2421H], AL
        HLT
```

INPUT:



OUTPUT :



EXPERIMENT - 15

SORTING IN DESCENDING ORDER:

Write a program to sort 16 bytes residing in locations starting from 0C20H / 2400H, in descending order in the same location.

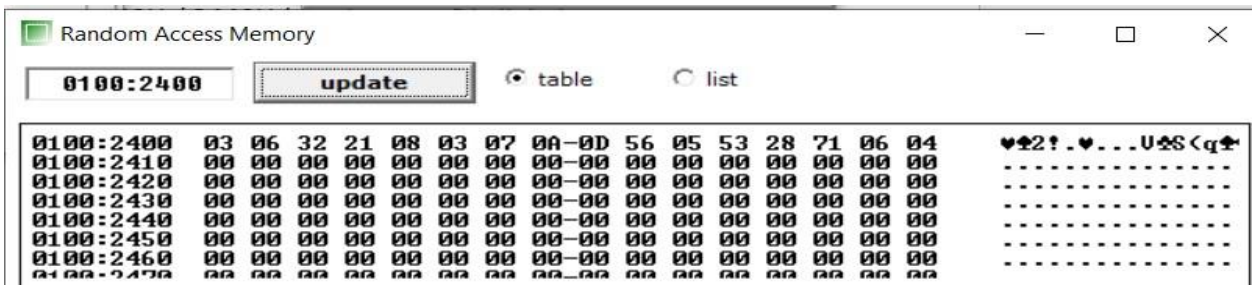
SOFTWARE USED: EMU8086 Simulator

CODE:

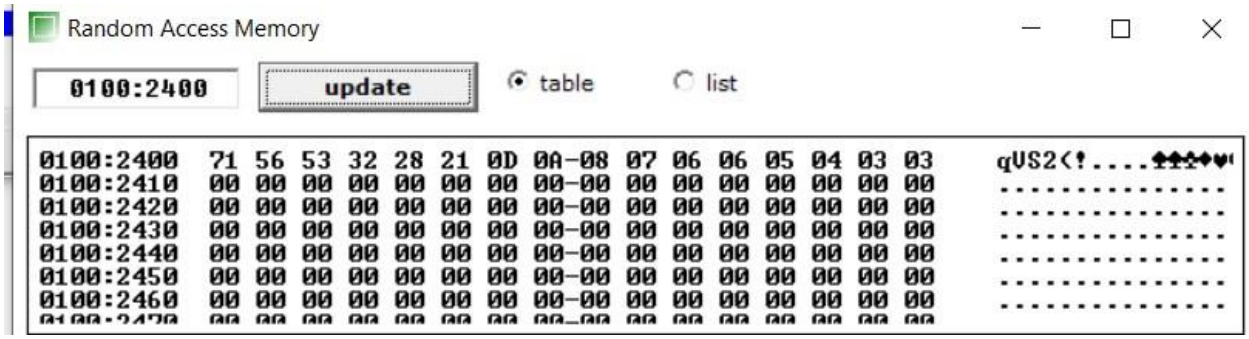
```
CODE:

START:  MOV SI , 2400H
        MOV BH, 10H
        MOV CL, 00H
CHECK:  MOV AL, [SI]
        INC SI
        MOV BL, [SI]
        CMP AL, BL
        JNC NEXTBYTE
        JZ  NEXTBYTE
        XCHG AL, [SI]
        DEC SI
        MOV [SI], AL
        INC SI
        MOV CL, 01H
NEXTBYTE: DEC BH
        JNZ CHECK
        CMP CL, 01H
        JZ  START
        HLT
```

INPUT:



OUTPUT :



EXPERIMENT NO : 16

ADDITION OF TWO 20-DIGIT BCD NO's:

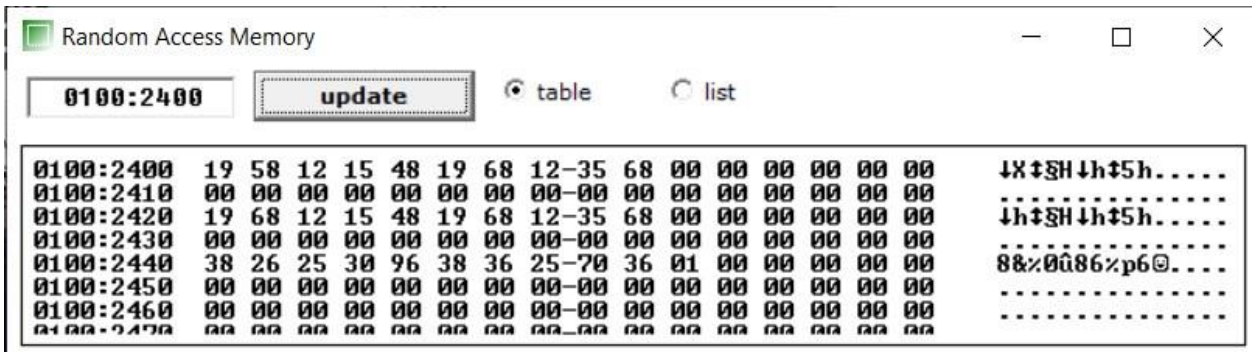
Write a program to add a 20-digit BCD number residing in location 0C20H / 2400H, with another 20-digit BCD number residing in location 0C30H / 2420H and store the result at location 0C40H / 2440H (result space 21 digits).

SOFTWARE USED: EMU8086 Simulator

CODE:

```
CODE:
        MOV  CL, 0AH
        MOV  SI, 2400H
        MOV  DI, 2420H
        MOV  BX, 2440H
NEXT:   MOV  AH,[SI]
        MOV  AL, [DI]
        ADC  AL, AH
        DAA
        MOV  [BX], AL
        INC  SI
        INC  DI
        INC  BX
        DEC  CL
        JNZ  NEXT
        MOV  AL,00H
        ADC  AL, 00H
        MOV  [BX], AL
        HLT
```

INPUT AND OUTPUT:



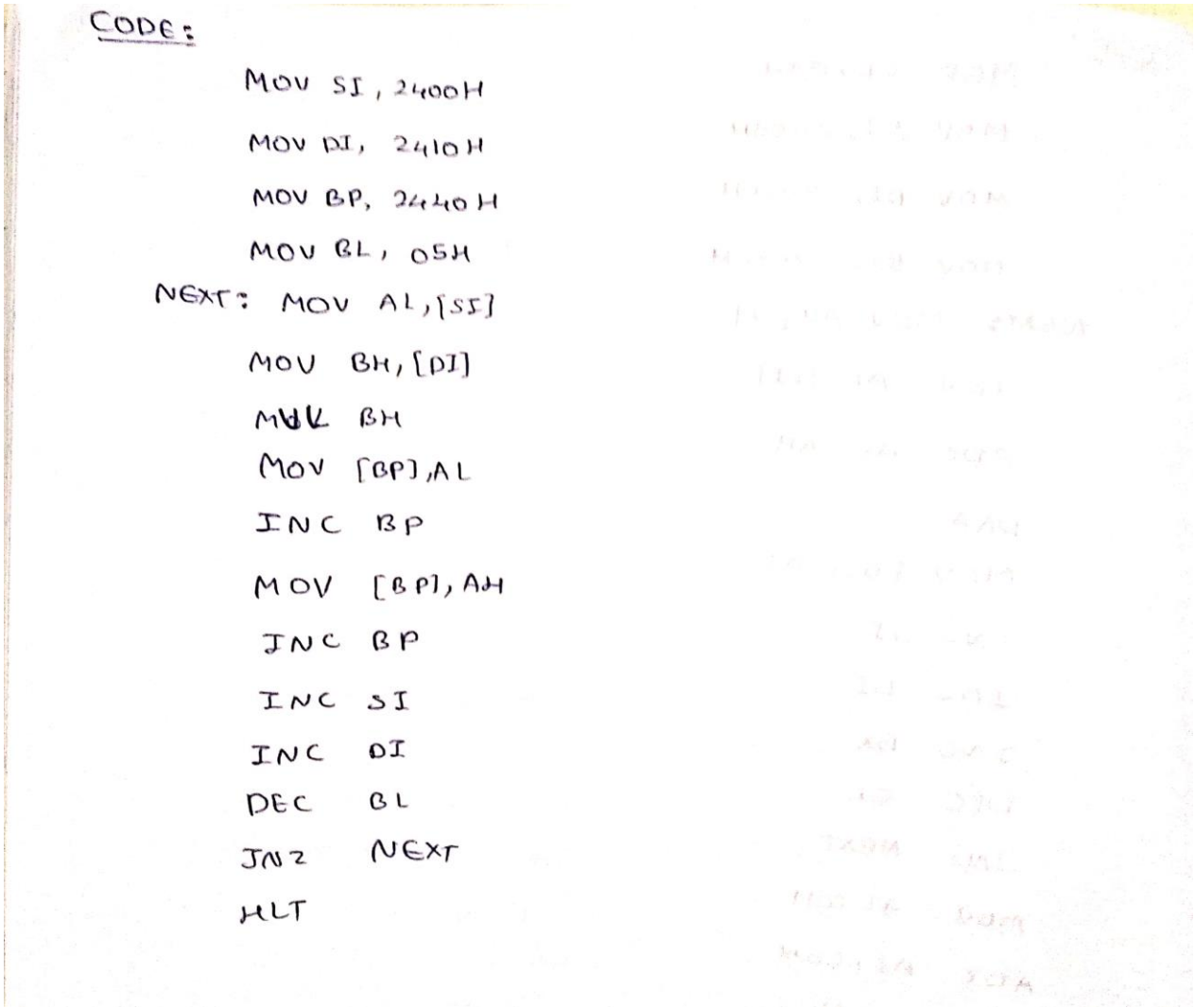
EXPERIMENT - 17

MULTIPLICATION OF TWO 5-BYTE BINARY NUMBERS:

Write a program to multiply a 5byte binary number residing in location at 0C20H / 2400H by another 5-byte number residing in location at 0C30H / 2410H and store the result at 0C40H / 2440H (result space 10 bytes).

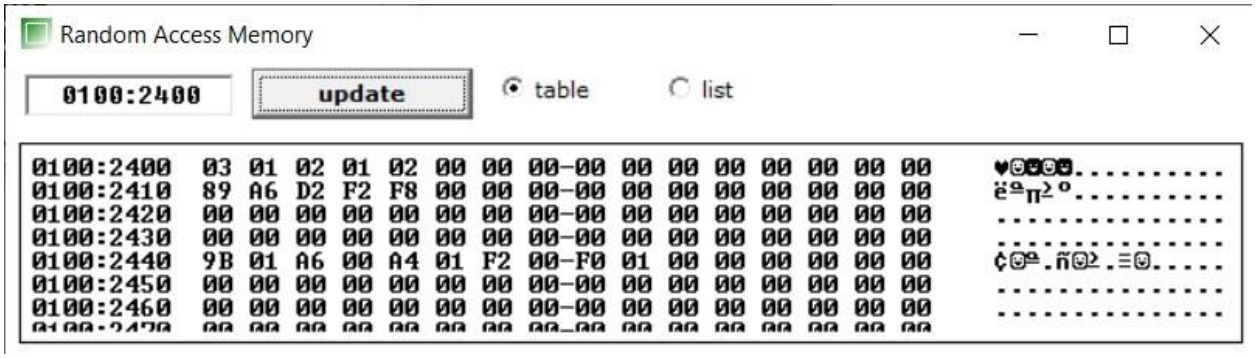
SOFTWARE USED :EMU8086 Simulator

CODE :



```
CODE:
        MOV SI, 2400H
        MOV DI, 2410H
        MOV BP, 2440H
        MOV BL, 05H
NEXT:   MOV AL,[SI]
        MOV BH,[DI]
        MUL BH
        MOV [BP],AL
        INC BP
        MOV [BP],AH
        INC BP
        INC SI
        INC DI
        DEC BL
        JNZ NEXT
        HLT
```

INPUT AND OUTPUT:



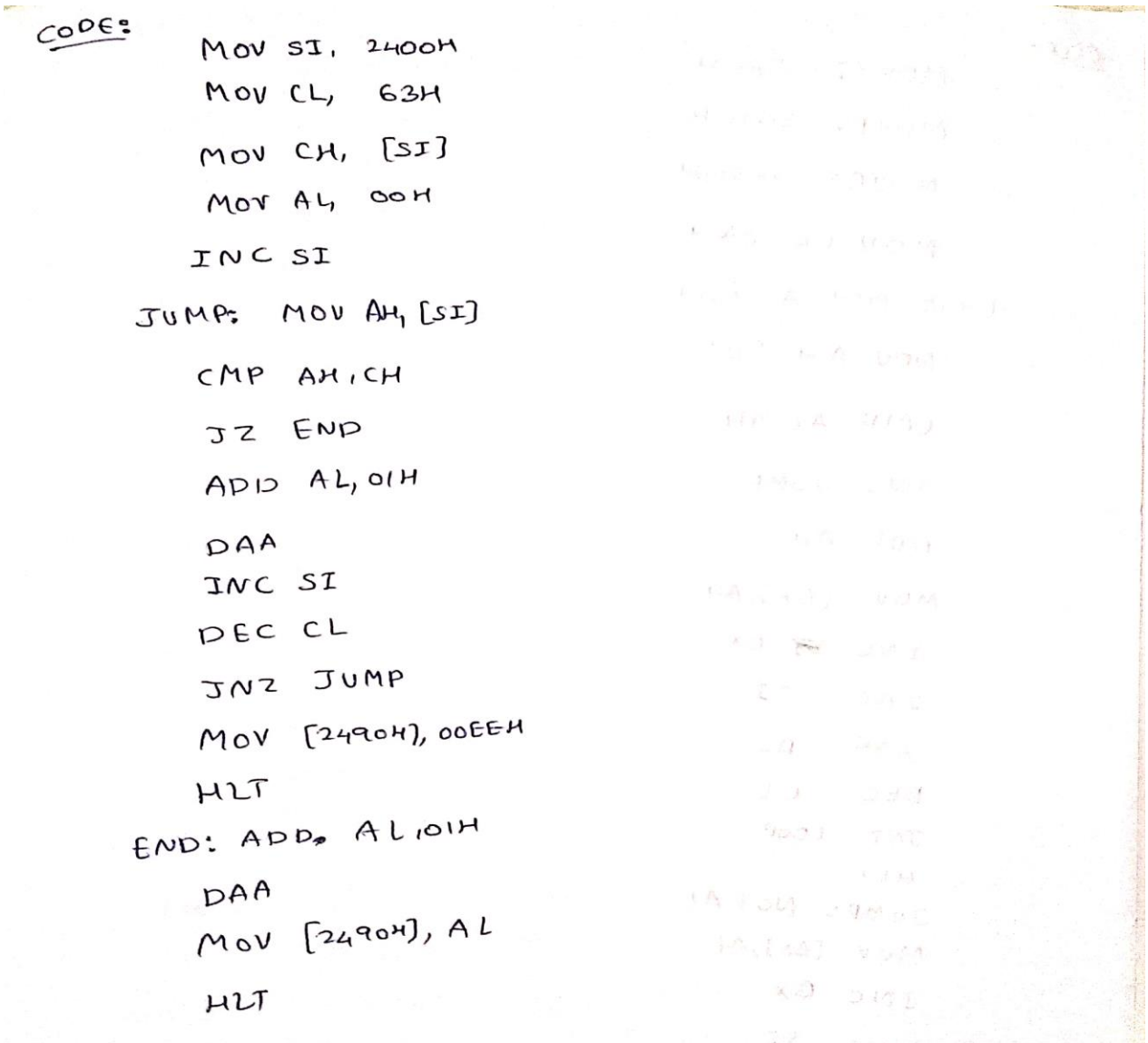
EXPERIMENT - 18

SIZE OF A BLOCK ENDING WITH A SPECIFIED BYTE:

Write a program to estimate the size of a block which starts at 0C21H / 2401H and ends up with a data specified at 0C20H / 2400H. Keep the count (in BCD) at 0C90H / 2490H. The specified byte is also a part of the block. If the specified byte is not encountered within 99 numbers, put EEH at 0C90H / 2490H.

SOFTWARE USED: EMU8086 Simulator

CODE:



```
CODE:
        MOV SI, 2400H
        MOV CL, 63H
        MOV CH, [SI]
        MOV AL, 00H
        INC SI
JUMP:   MOV AH, [SI]
        CMP AH, CH
        JZ END
        ADD AL, 01H
        DAA
        INC SI
        DEC CL
        JNZ JUMP
        MOV [2490H], 00EEH
        HLT
END:    ADD AL, 01H
        DAA
        MOV [2490H], AL
        HLT
```

INPUT:

OUTPUT :

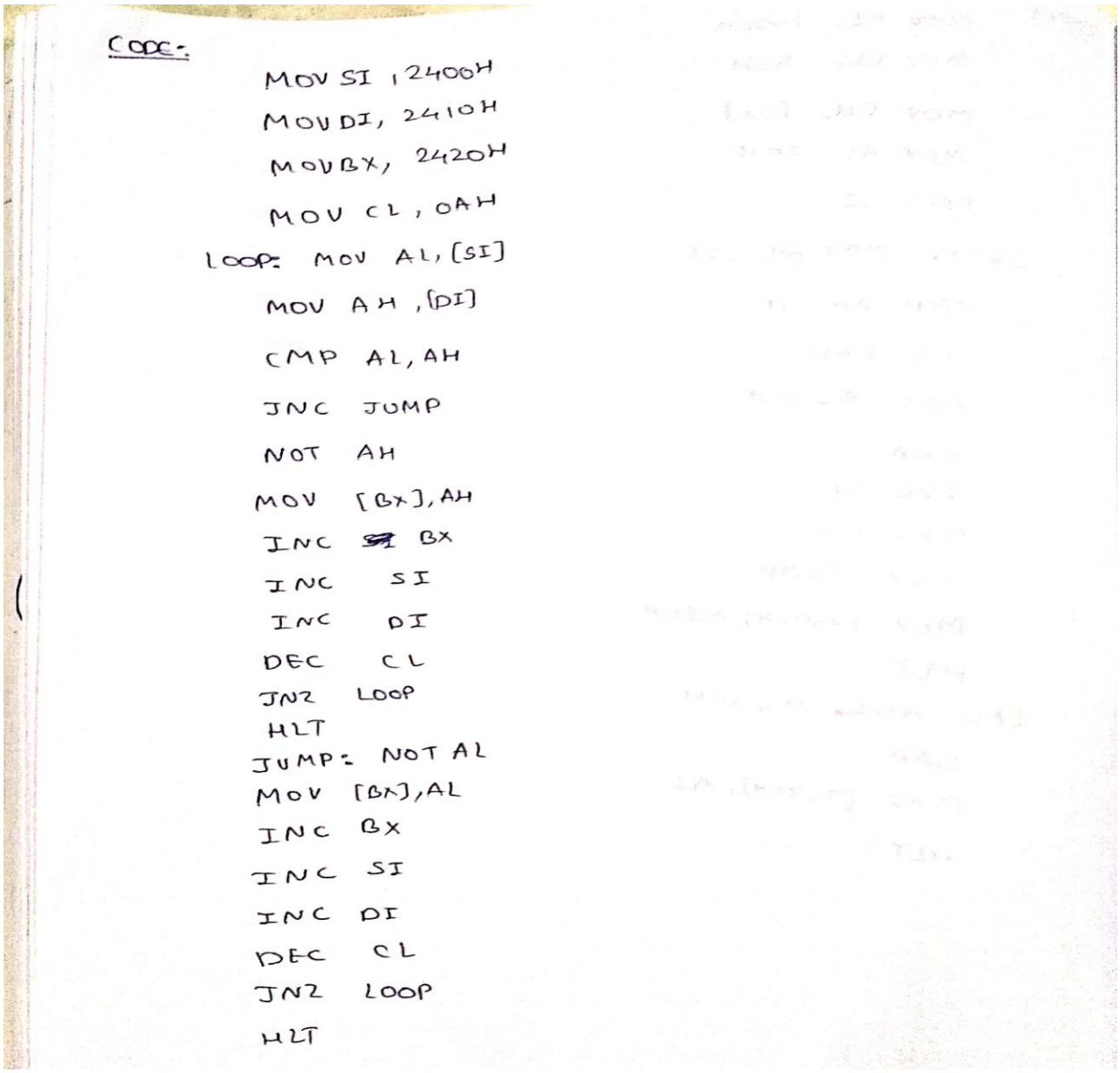
EXPERIMENT - 19

FORMATION OF A THIRD BLOCK:

Two blocks of 10 bytes each are residing locations 0C20H / 2400H and 0C30H / 2410H respectively, Write a program to form a third block of 10 bytes at locations starting from 0C40H / 2420H, such that each entry in that block is the one's complement of the larger of the corresponding entries of the first two blocks (the nth entry in the third block shall of the nth entry of the first block and the nth entry of the second block).

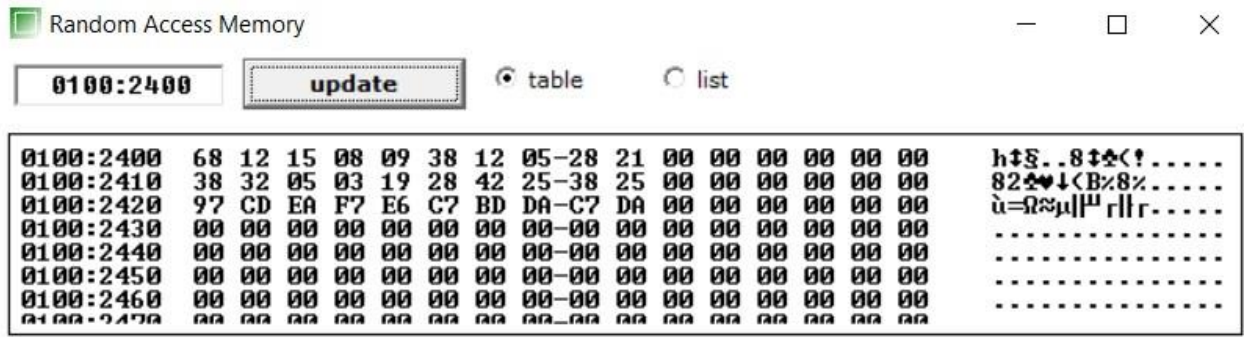
SOFTWARE USED: EMU8086 Simulator

CODE:



```
CODE:
        MOV SI , 2400H
        MOV DI, 2410H
        MOV BX, 2420H
        MOV CL , 0AH
LOOP:   MOV AL, [SI]
        MOV AH , [DI]
        CMP AL, AH
        JNC JUMP
        NOT AH
        MOV [BX], AH
        INC BX
        INC SI
        INC DI
        DEC CL
        JNZ LOOP
        HLT
JUMP:   NOT AL
        MOV [BX], AL
        INC BX
        INC SI
        INC DI
        DEC CL
        JNZ LOOP
        HLT
```

INPUT AND OUTPUT :



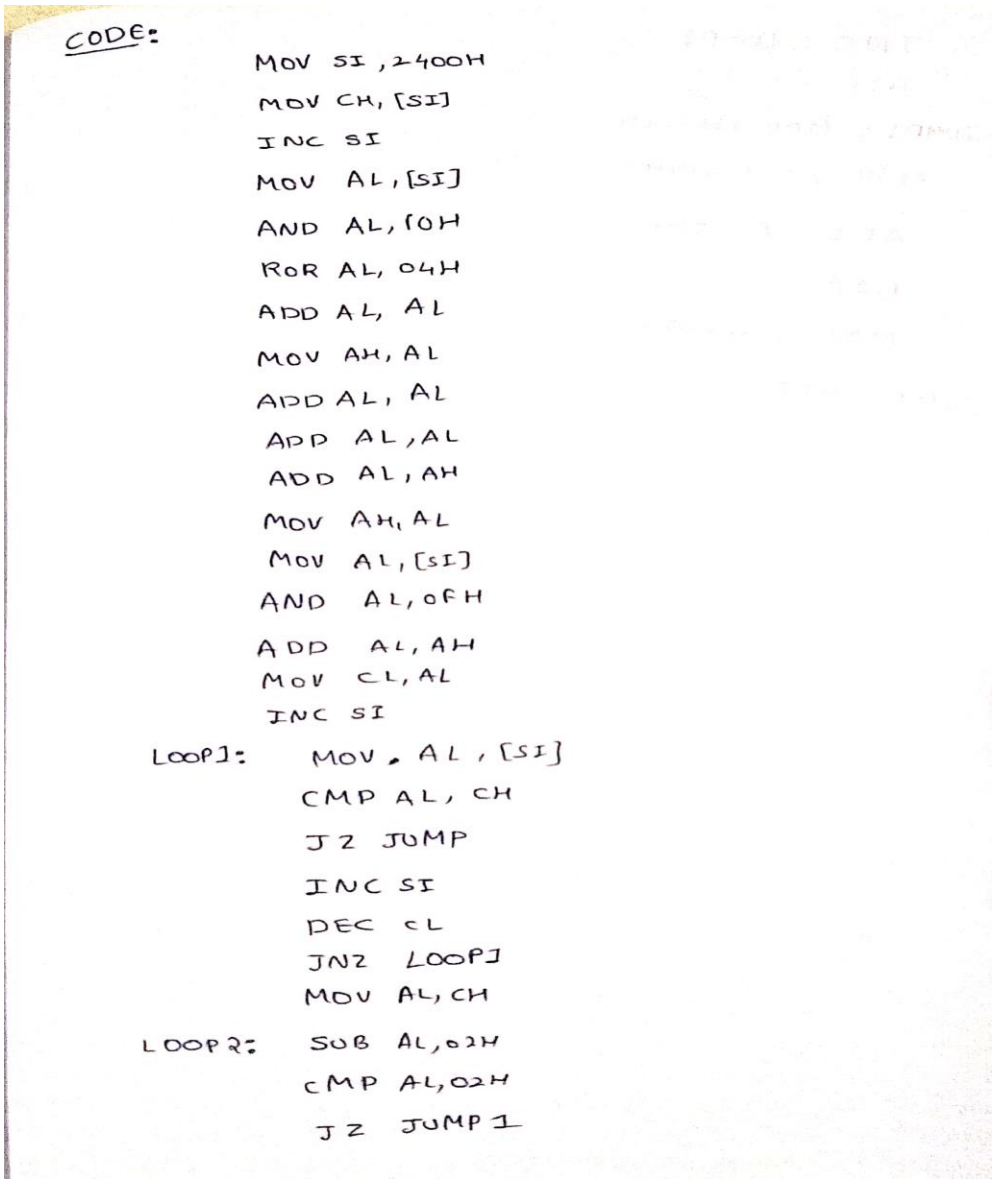
EXPERIMENT-20

INSERTION TO A LIST:

Write a program to insert the content at the location 0C20H / 2400H to a list which begins at 0C22H / 2402H, if not already present in the list and only if byte at location 0C20H / 2400H is divisible by two. The number of bytes in the list is given at location 0C21H / 2401H (in BCD). If the insertion is valid, insert the byte at the end of the list and modify the count accordingly.

SOFTWARE USED: EMU8086 Simulator

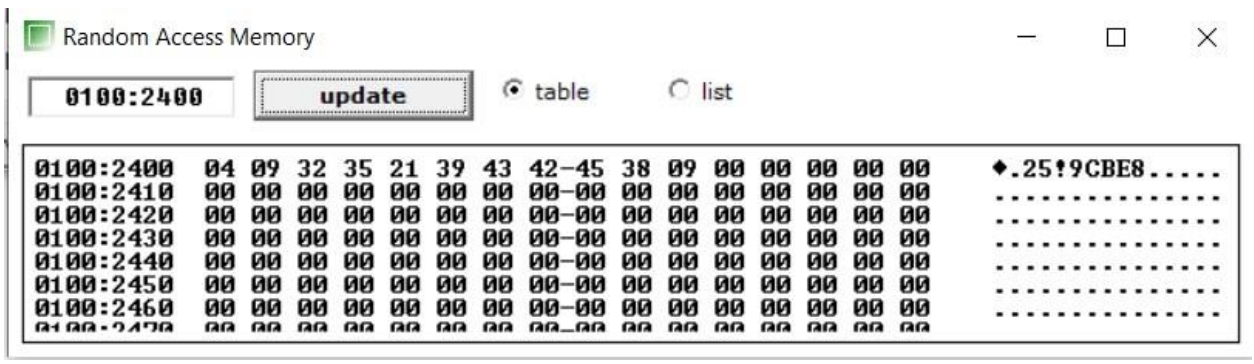
CODE:



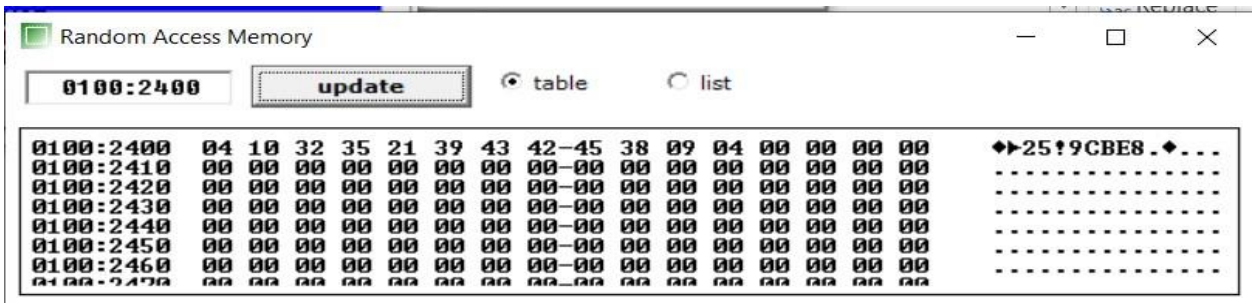
```
CODE:
        MOV SI,2400H
        MOV CH,[SI]
        INC SI
        MOV AL,[SI]
        AND AL,10H
        ROR AL,04H
        ADD AL,AL
        MOV AH,AL
        ADD AL,AL
        ADD AL,AL
        ADD AL,AH
        MOV AH,AL
        MOV AL,[SI]
        AND AL,0FH
        ADD AL,AH
        MOV CL,AL
        INC SI
LOOP1:  MOV AL,[SI]
        CMP AL,CH
        JZ JUMP
        INC SI
        DEC CL
        JNZ LOOP1
        MOV AL,CH
LOOP2:  SUB AL,02H
        CMP AL,02H
        JZ JUMP1
```

```
        JNC   LOOP2
        HLT
JUMP1 : MOV [SI], CH
        MOV  AL, [2401H]
        ADD  AL, 01H
        DAA
        MOV  [2401H], AL
JUMP : HLT
```

INPUT :



OUTPUT :


Random Access Memory
0100:2400
update
table
list
0100:2400 04 10 32 35 21 39 43 42-45 38 09 04 00 00 00 00 ♦►25!9CBE8.♦...
0100:2410 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:2420 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:2430 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:2440 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:2450 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:2460 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....