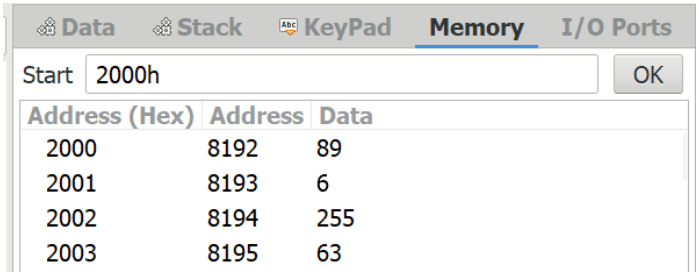
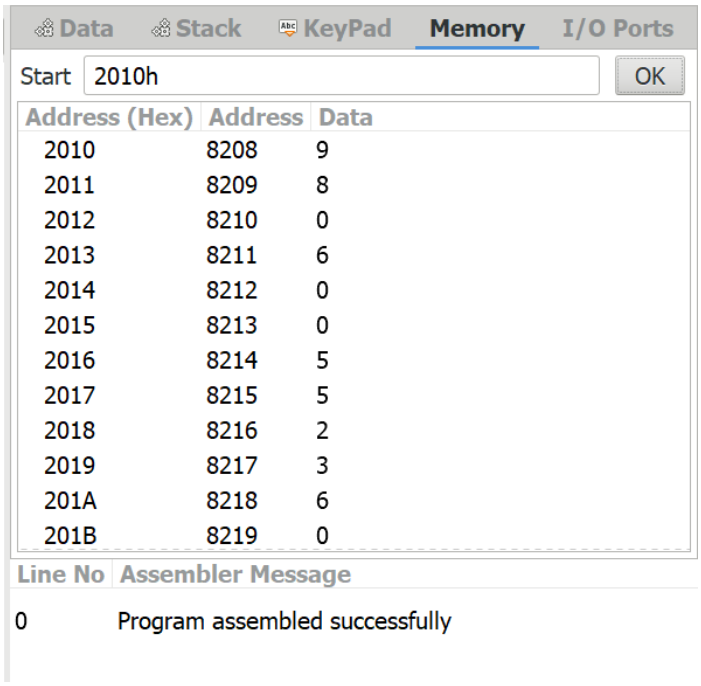
MIT ASSIGNMENT – 8

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.**

```
;Program1
MVI C,04H ; COUNTER
LXI D,2010H ; DESTINATION
LXI H,2000H ; SOURCE
LOOP: MOV A,M
CALL bin_bcd
DCR C
INX H
JNZ LOOP
HLT

bin_bcd: PUSH H
MVI H,0H ; HUNDREDS
MVI L,0H ; TENS

S1: CPI 64H
JC S2
SUI 64H
INR H
JMP S1

S2: CPI 0AH
JC S3
SUI 0AH
INR L
JMP S2

S3: STAX D ; STORE 1'S POSITION
INX D
MOV A,L
STAX D ; STORE 10'S POSITION
INX D
MOV A,H
```

```
STAX D ; STORE 100'S POSITION
INX D
POP H
RET
```

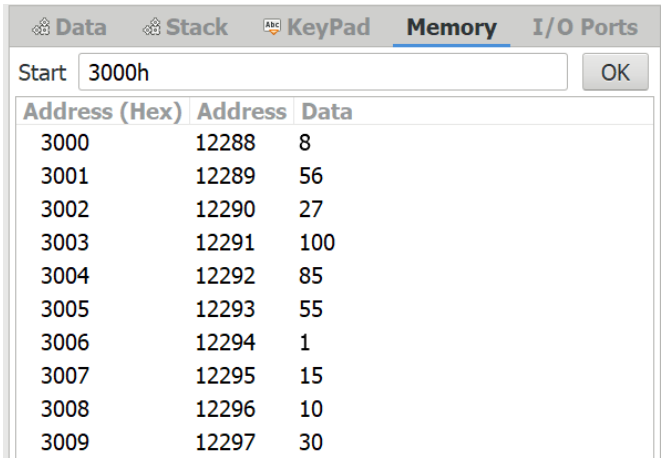




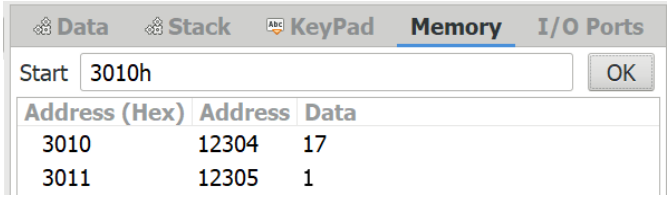
2. **A set of ten BCD readings is stored in the Input Buffer. Convert the numbers into binary and add the numbers. Store the sum in the Output Buffer, the sum can be larger than FFH.**

```
;Program2
MVI B,0AH ;COUNTER
LXI D,3000H ; SOURCE POINTER
LXI H,0000H ; SUM REGISTER
LOOP: CALL BCD2BIN
MOV C,A
MOV A,B
MVI B,00H
DAD B ; ADDING THE BINARY EQUIVIVALENT OF BCD
INX D
MOV B,A
DCR B
JNZ LOOP
SHLD 3010H ; STORING ANSWER
```

```
HLT

BCD2BIN: PUSH B
PUSH H
LDAX D
ANI 0FH ; UNPACKING BCD
MOV B,A
LDAX D
ANI 0F0H
RRC
RRC
RRC
RRC
MOV H,A
MVI C,09H
MUL10: ADD H ; A*10+B
DCR C
JNZ MUL10
ADD B
POP H
POP B
RET
```



Address (Hex)	Address	Data
3000	12288	8
3001	12289	56
3002	12290	27
3003	12291	100
3004	12292	85
3005	12293	55
3006	12294	1
3007	12295	15
3008	12296	10
3009	12297	30



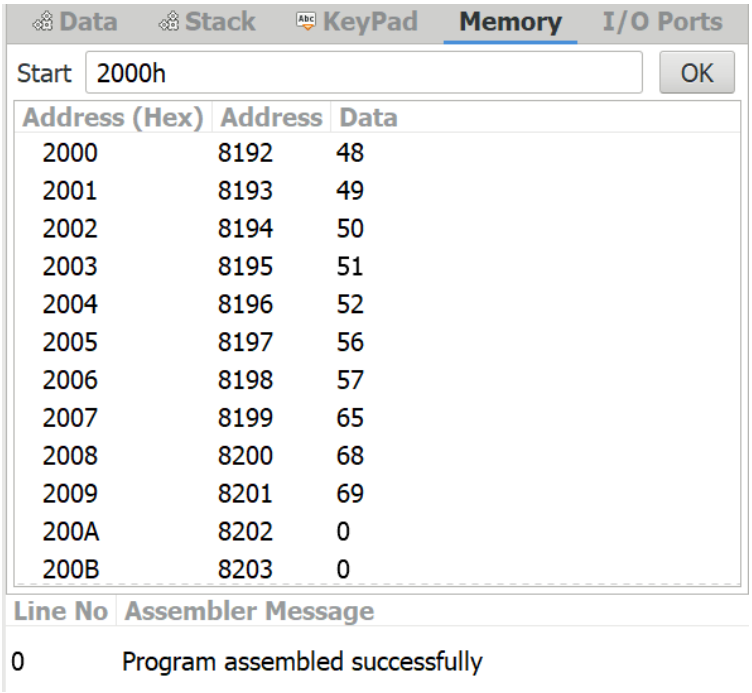
Address (Hex)	Address	Data
3010	12304	17
3011	12305	1

Output: 1 and 17 = (0000 0001 0001 0001) which is $(273)_{10}$

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.**

```
;Program3
```

```
MVI B,0AH;COUNTER
MVI C,00H ;SUM REGISTER
LXI D,2000H ;SOURCE POINTER FOR ASCIILXI H,2010H
LXI H,2010H ;DESTINATION POINTER FOR EQUIVIVALENT BINARY

LOOP: LDAX D
CALL ASCBIN
MOV M,A ;STORING THE EQUIVIVALENT
MOV A,C
ADD M
MOV C,A ;ADDING
INX H
INX D
DCR B
JNZ LOOP
MOV M,C ;STORING SUM AT THE END
HLT

;INPUT A - ASCII CODE
;OUTPUT A - BINARY EQUIVIVALENT
ASCBIN: SUI 30H
CPI 0AH ;IF LESS THAN 10 RETURN
RC
SUI 07H ;FOR A-F
RET
```



Data | Stack | KeyPad | **Memory** | I/O Ports

Start 2010h OK

Address (Hex)	Address	Data
2010	8208	0
2011	8209	1
2012	8210	2
2013	8211	3
2014	8212	4
2015	8213	8
2016	8214	9
2017	8215	10
2018	8216	13
2019	8217	14
201A	8218	64
201B	8219	0

Line No	Assembler Message
0	Program assembled successfully