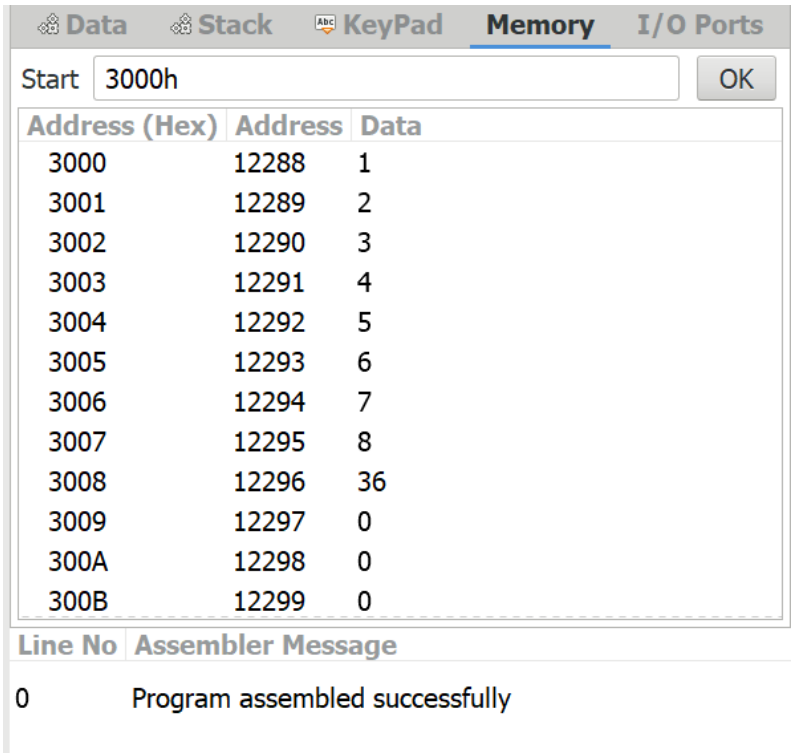
MIT ASSIGNMENT – 5

1. **Write a program to Add ten 8-bit numbers stored in memory starting from 3000H. Store your result at 3050H.**

```
;Program 1
;Adding numbers from 3000h to 3009h
lxi h,3000h
mvi c,08h
mvi a,00h
mvi d,00h
loop: add m
inx h
jnc noCarry
inr d
noCarry: dcr c
jnz loop
mov e,a
xchg
shld 3008h
hlt
```



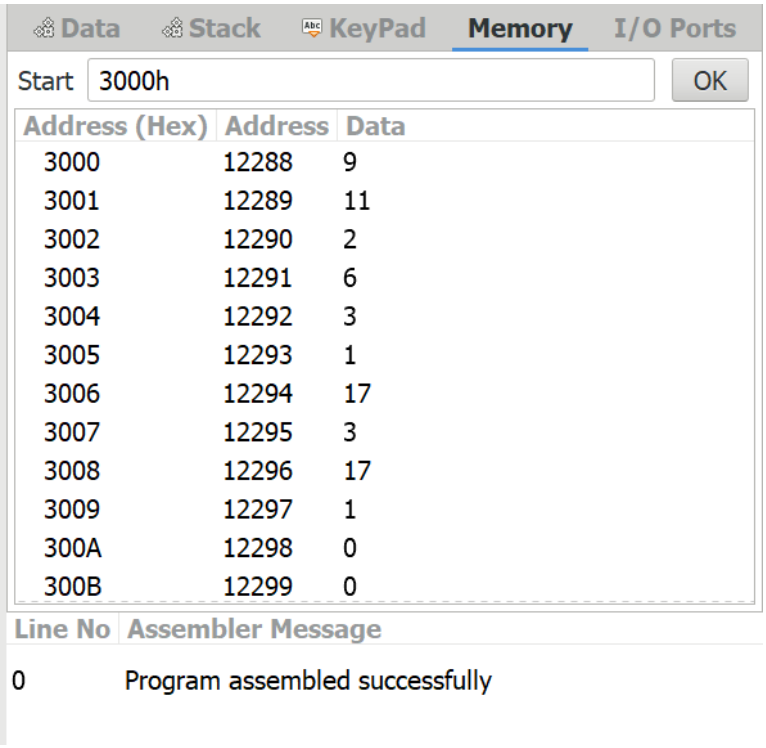
2. **Write a program to find smallest/largest number from the array of 8-bit data. Assume suitable memory location for data and result.**

```
;Program2
;find smallest/largest number from the array
```

```
mvi c,07h
lxi h,3000h
mov a,m
call largest
sta 3008h
mvi c,07h
lxi h,3000h
mov a,m
call smallest
sta 3009h
hlt


largest: inx h
cmp m
jnc noCarry
mov a,m
noCarry: dcr c
jz zero
call largest
zero: ret

smallest: inx h
cmp m
jc Carry
mov a,m
Carry: dcr c
jz zer
call smallest
zer: ret
```

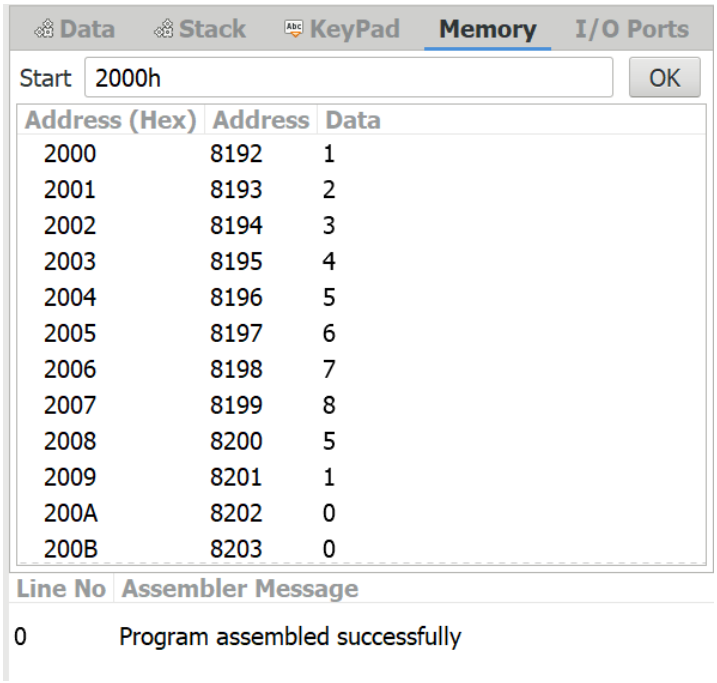


(Smallest at 3009h and largest at 3008h)

3. **Write a Program to search an 8-bit number from the array of 8-bit data.**

```
;Program3
;Number to find is stored at address 2008h
lxi h,2000h
lda 2008h
mvi c,08h
mvi b,00h
loop: cmp m
jnz next
mvi b,01h
jmp exit
next: inx h
dcr c
jz exit
jmp loop

exit: mov a,b
sta 2009h
hlt
```

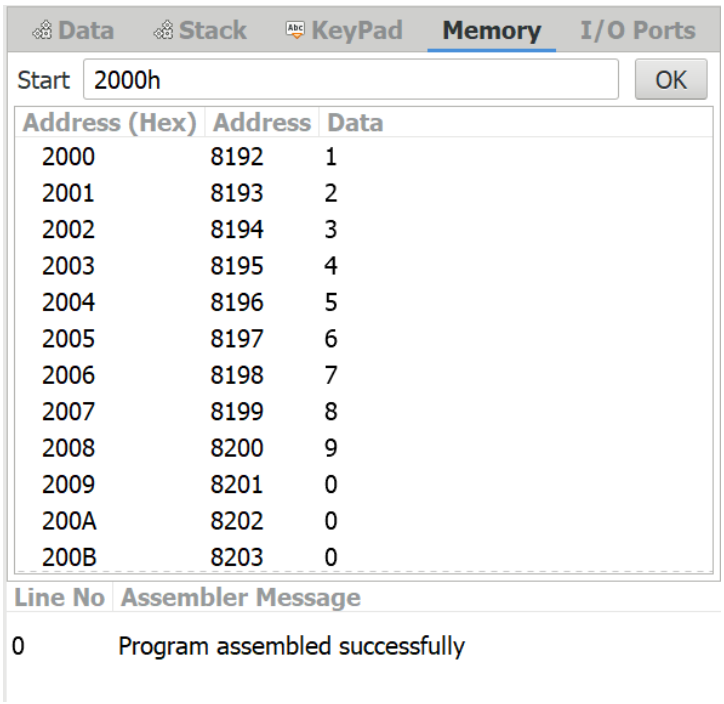


Data | Stack | KeyPad | Memory | I/O Ports

Start 2000h OK

Address (Hex)	Address	Data
2000	8192	1
2001	8193	2
2002	8194	3
2003	8195	4
2004	8196	5
2005	8197	6
2006	8198	7
2007	8199	8
2008	8200	5
2009	8201	1
200A	8202	0
200B	8203	0

Line No	Assembler Message
0	Program assembled successfully



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

Start 2000h OK

Address (Hex)	Address	Data
2000	8192	1
2001	8193	2
2002	8194	3
2003	8195	4
2004	8196	5
2005	8197	6
2006	8198	7
2007	8199	8
2008	8200	9
2009	8201	0
200A	8202	0
200B	8203	0

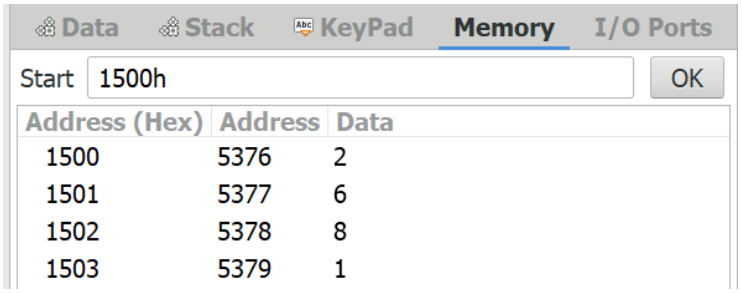
Line No	Assembler Message
0	Program assembled successfully

If number is found, 1 will be stored at address 2009h and if not found 0 will be stored there. Number to find is present at address 2008h. Array elements are from: 2000h to 2007h

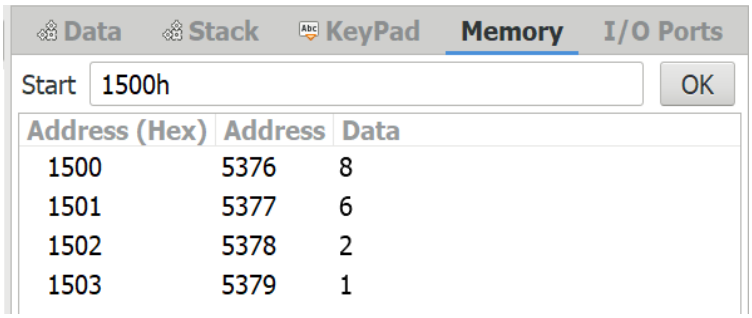
4. **Write a Program to arrange data in ascending/descending order. Assume suitable memory location for data and result.**

```
;Problem 4
;DESCENDING
MVI B,04H
start: LXI H,1500h
MVI C,03H
back: MOV A,M
```

```
INX H
CMP M
JNC skip
MOV D,M
MOV M,A
DCX H
MOV M,D
INX H
skip: DCR C
JNZ back
DCR B
JNZ start
HLT
```



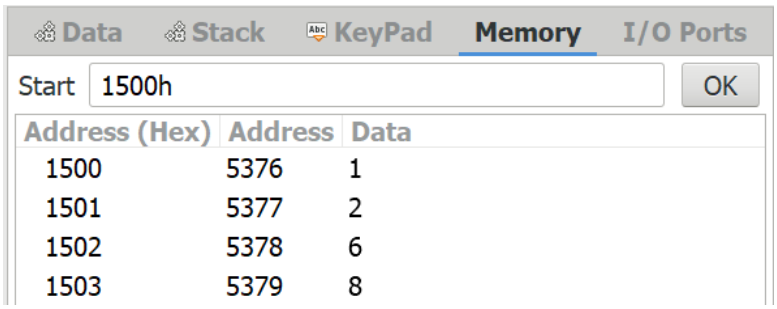
(Before sorting)



(After sorting)

```
;Problem 4
;ASCENDING
MVI B,04H
start: LXI H,1500h
MVI C,03H
back: MOV A,M
INX H
CMP M
JC skip
MOV D,M
MOV M,A
DCX H
```

```
MOV M,D
INX H
skip: DCR C
JNZ back
DCR B
JNZ start
HLT
```



(After sorting the previous output in ascending order)