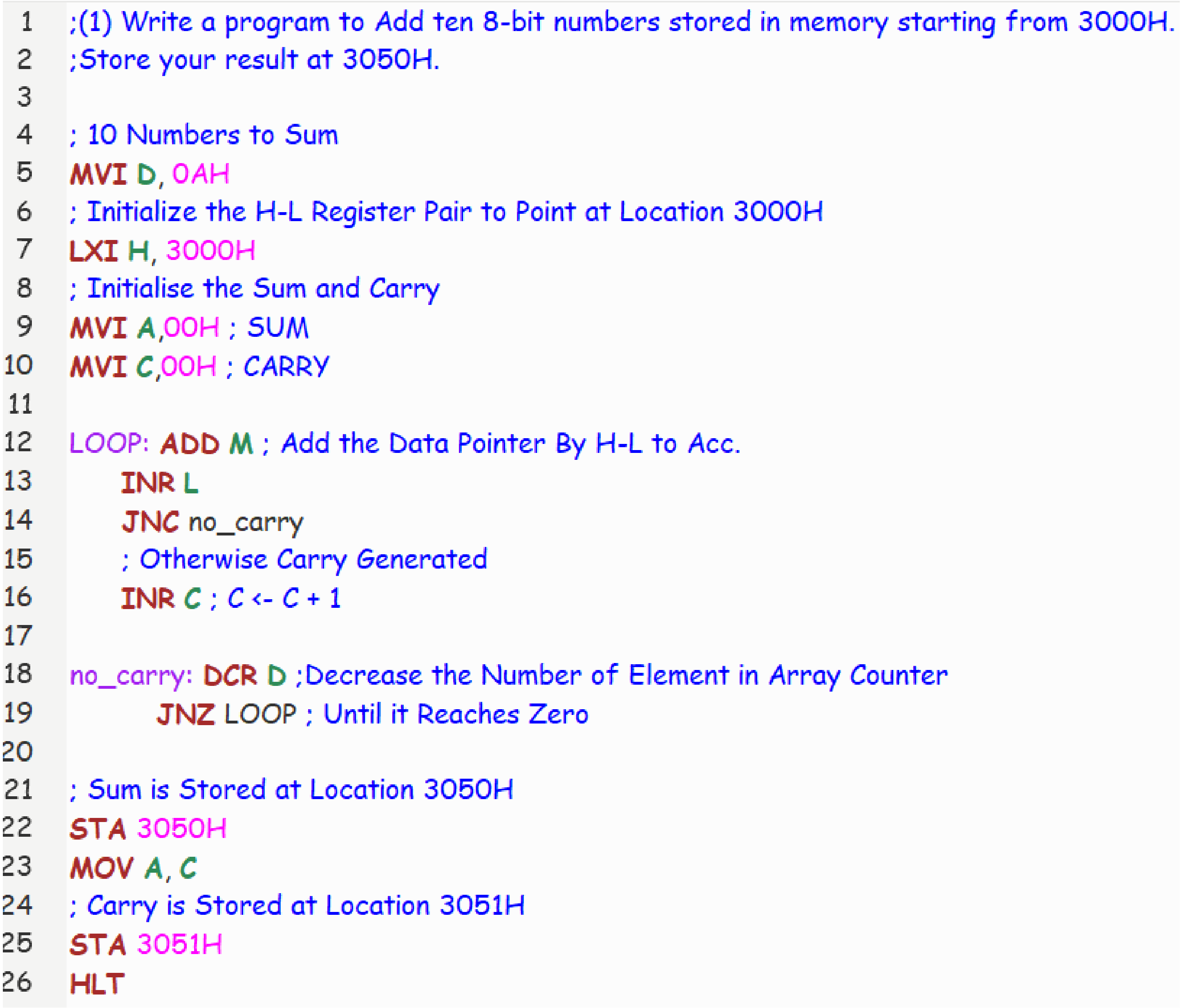
M.I.T. LAB Assignment – 05

U19CS012

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

Notepad Code:



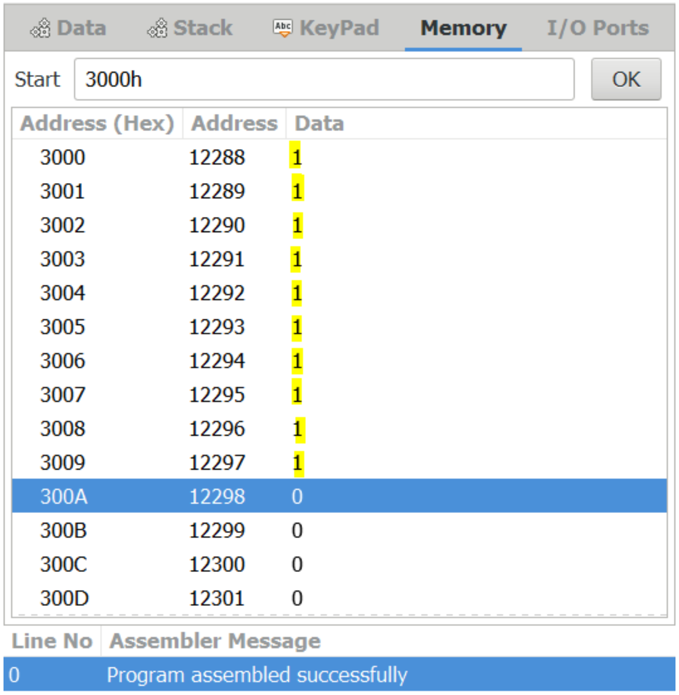
```
;(1) Write a program to Add ten 8-bit numbers stored in memory starting from 3000H.
;Store your result at 3050H.

; 10 Numbers to Sum
MVI D, 0AH
; Initialize the H-L Register Pair to Point at Location 3000H
LXI H, 3000H
; Initialise the Sum and Carry
MVI A,00H ; SUM
MVI C,00H ; CARRY

LOOP: ADD M ; Add the Data Pointer By H-L to Acc.
      INR L
      JNC no_carry
      ; Otherwise Carry Generated
      INR C ; C <- C + 1

no_carry: DCR D ;Decrease the Number of Element in Array Counter
          JNZ LOOP ; Until it Reaches Zero

; Sum is Stored at Location 3050H
STA 3050H
MOV A, C
; Carry is Stored at Location 3051H
STA 3051H
HLT
```

Registers and Memory:

Eg : (1) Array = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Sum of Array Elements = 10 = (0A)H

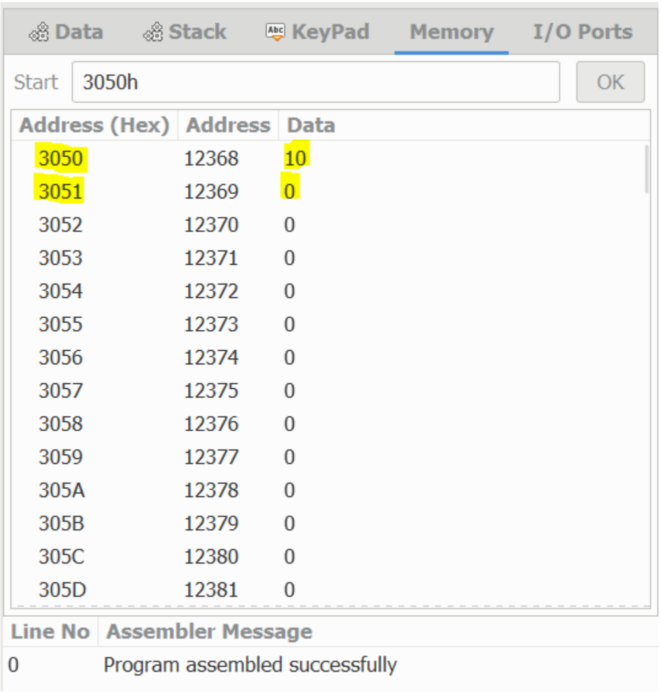
Sum = (0A)H = 10 [Location 3050H] & Carry = (00)H = 0 [Location 3051H]



Start 3000h

Address (Hex)	Address	Data
3000	12288	1
3001	12289	1
3002	12290	1
3003	12291	1
3004	12292	1
3005	12293	1
3006	12294	1
3007	12295	1
3008	12296	1
3009	12297	1
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully



Start 3050h

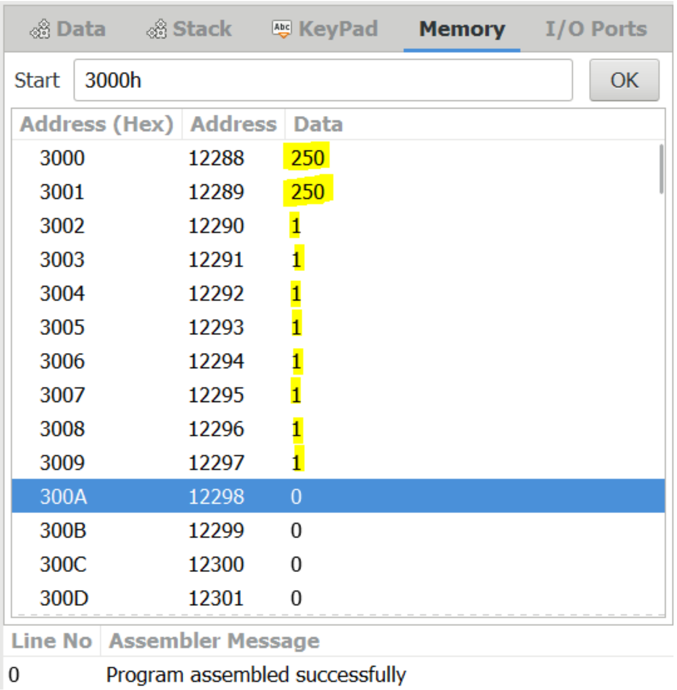
Address (Hex)	Address	Data
3050	12368	10
3051	12369	0
3052	12370	0
3053	12371	0
3054	12372	0
3055	12373	0
3056	12374	0
3057	12375	0
3058	12376	0
3059	12377	0
305A	12378	0
305B	12379	0
305C	12380	0
305D	12381	0

Line No	Assembler Message
0	Program assembled successfully

(2) Array = [250, 250, 1, 1, 1, 1, 1, 1, 1, 1]

Sum of Array Elements = 508 = (1FC)H

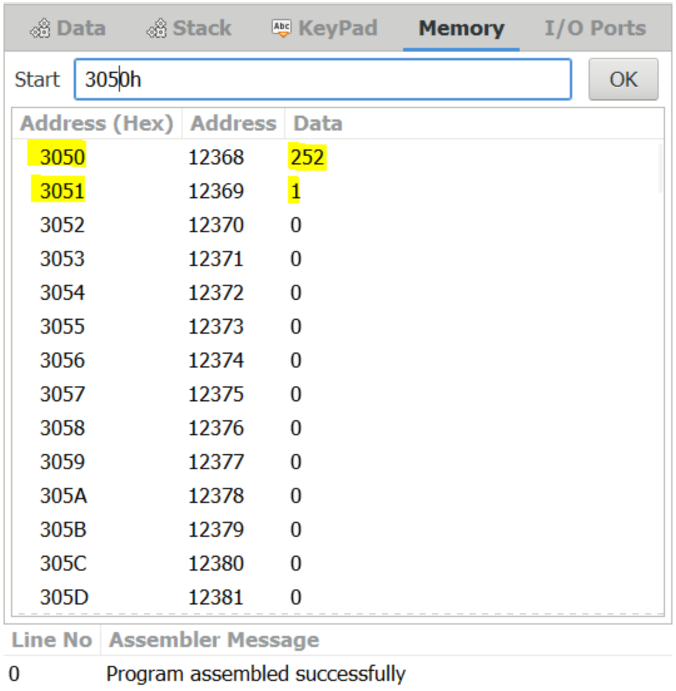
Sum = (FC)H = 252 [Location 3050H] & Carry = (01)H = 1 [Location 3051H]



Start 3000h

Address (Hex)	Address	Data
3000	12288	250
3001	12289	250
3002	12290	1
3003	12291	1
3004	12292	1
3005	12293	1
3006	12294	1
3007	12295	1
3008	12296	1
3009	12297	1
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully



Start 3050h

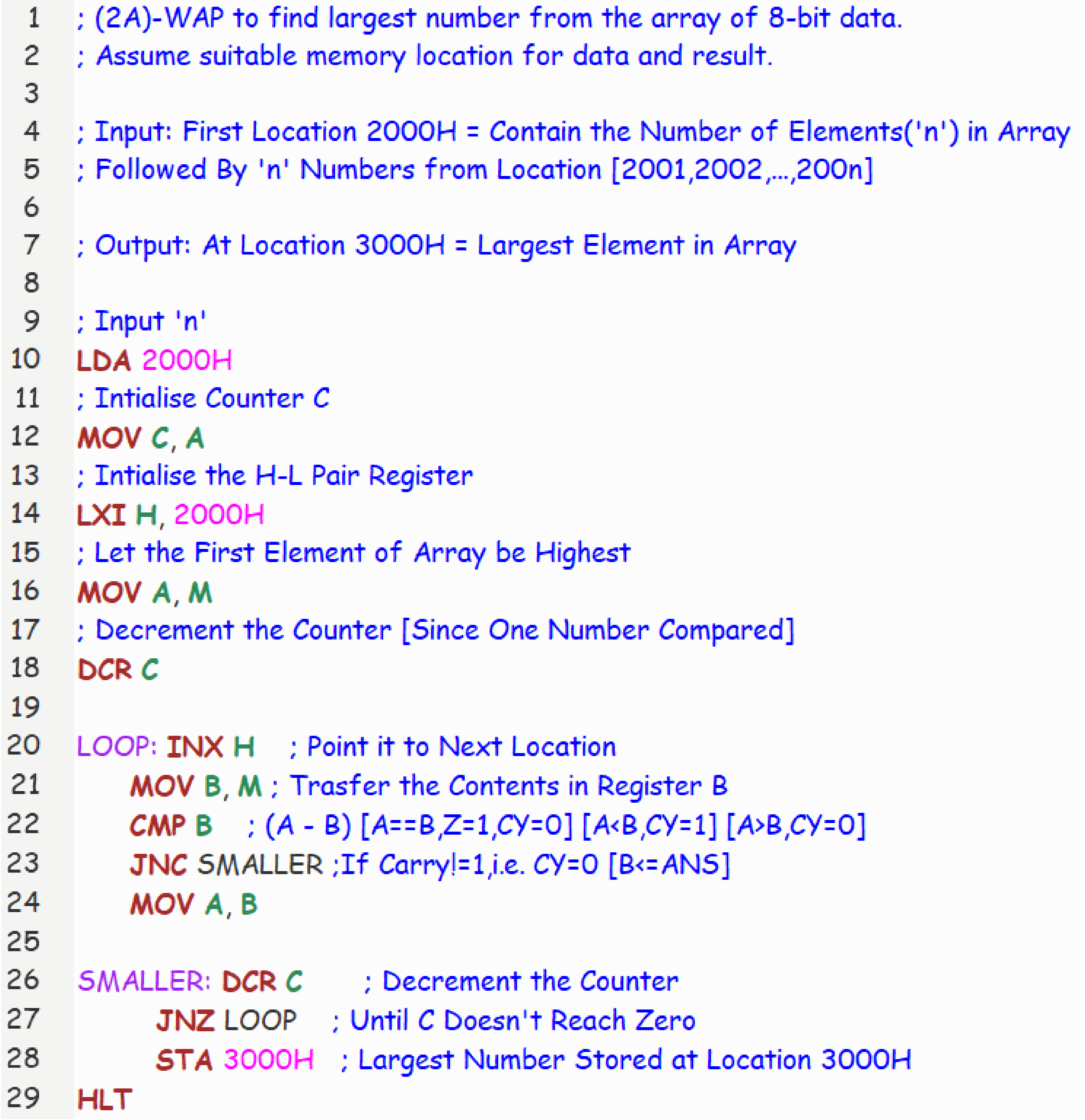
Address (Hex)	Address	Data
3050	12368	252
3051	12369	1
3052	12370	0
3053	12371	0
3054	12372	0
3055	12373	0
3056	12374	0
3057	12375	0
3058	12376	0
3059	12377	0
305A	12378	0
305B	12379	0
305C	12380	0
305D	12381	0

Line No	Assembler Message
0	Program assembled successfully

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

Notepad Code:

(A) Largest Element in Array

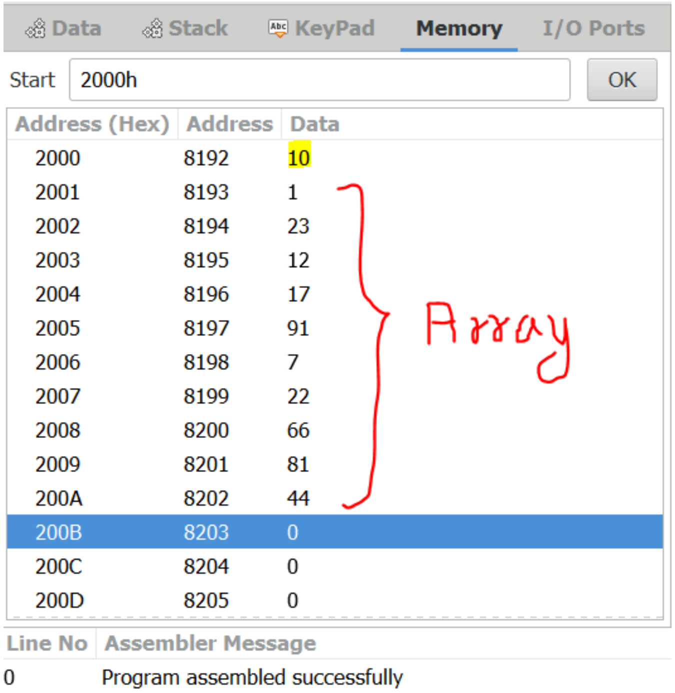


```
; (2A)-WAP to find largest number from the array of 8-bit data.
; Assume suitable memory location for data and result.

; Input: First Location 2000H = Contain the Number of Elements('n') in Array
; Followed By 'n' Numbers from Location [2001,2002,...,200n]

; Output: At Location 3000H = Largest Element in Array

; Input 'n'
LDA 2000H
; Intialise Counter C
MOV C, A
; Intialise the H-L Pair Register
LXI H, 2000H
; Let the First Element of Array be Highest
MOV A, M
; Decrement the Counter [Since One Number Compared]
DCR C

LOOP: INX H    ; Point it to Next Location
    MOV B, M ; Trasfer the Contents in Register B
    CMP B    ; (A - B) [A==B,Z=1,CY=0] [A<B,CY=1] [A>B,CY=0]
    JNC SMALLER ;If Carry!=1,i.e. CY=0 [B<=ANS]
    MOV A, B

SMALLER: DCR C      ; Decrement the Counter
      JNZ LOOP    ; Until C Doesn't Reach Zero
      STA 3000H   ; Largest Number Stored at Location 3000H
HLT
```

Registers and Memory [Largest Number Code]:

Eg : Array of 10 Numbers = [1, 23, 12, 17, 91, 7, 22, 66, 81, 44]

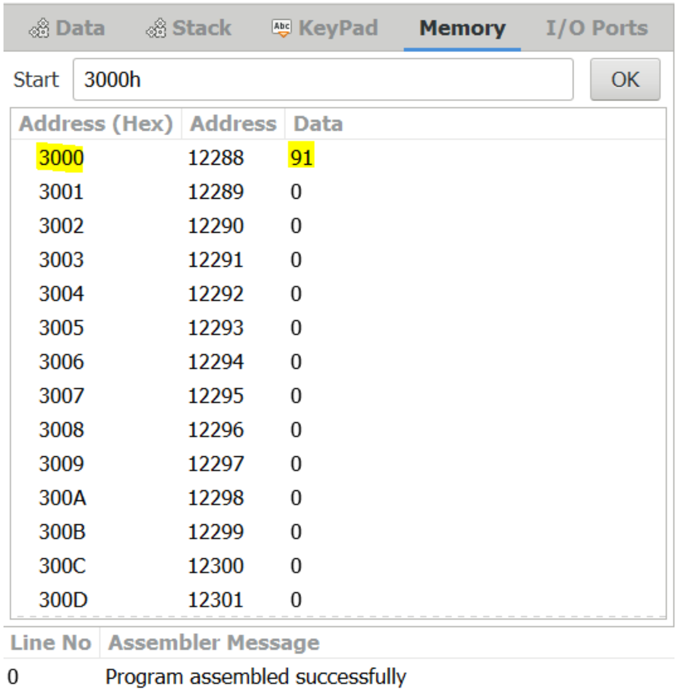
Largest Element in Array = 91



Start 2000h

Address (Hex)	Address	Data
2000	8192	10
2001	8193	1
2002	8194	23
2003	8195	12
2004	8196	17
2005	8197	91
2006	8198	7
2007	8199	22
2008	8200	66
2009	8201	81
200A	8202	44
200B	8203	0
200C	8204	0
200D	8205	0

Line No	Assembler Message
0	Program assembled successfully



Start 3000h

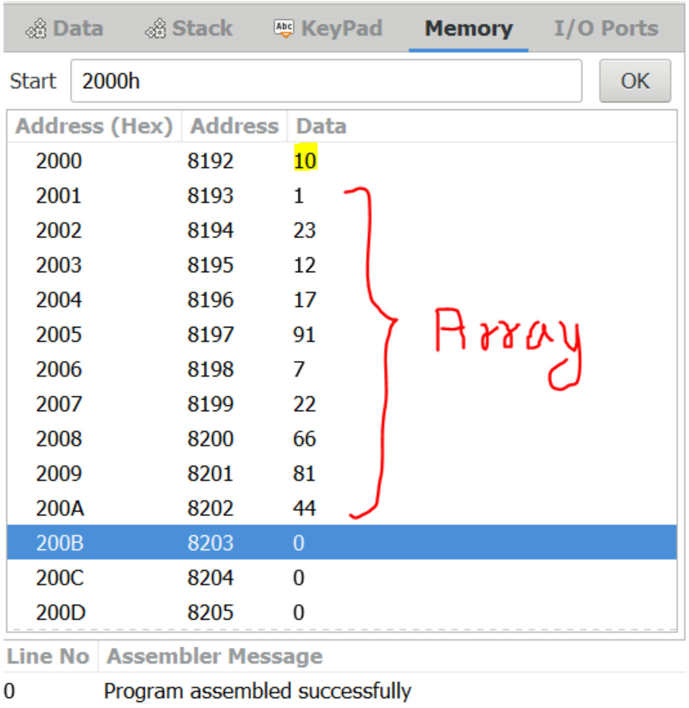
Address (Hex)	Address	Data
3000	12288	91
3001	12289	0
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully

Registers and Memory [Smallest Number Code]:

Eg : Array of 10 Numbers = [1, 23, 12, 17, 91, 7, 22, 66, 81, 44]

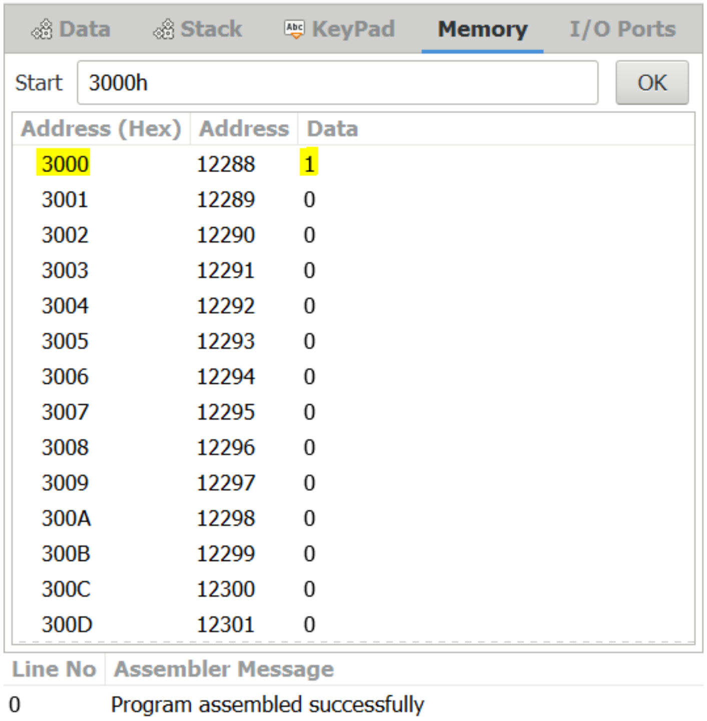
Smallest Element in Array = 1



Start 2000h

Address (Hex)	Address	Data
2000	8192	10
2001	8193	1
2002	8194	23
2003	8195	12
2004	8196	17
2005	8197	91
2006	8198	7
2007	8199	22
2008	8200	66
2009	8201	81
200A	8202	44
200B	8203	0
200C	8204	0
200D	8205	0

Line No	Assembler Message
0	Program assembled successfully

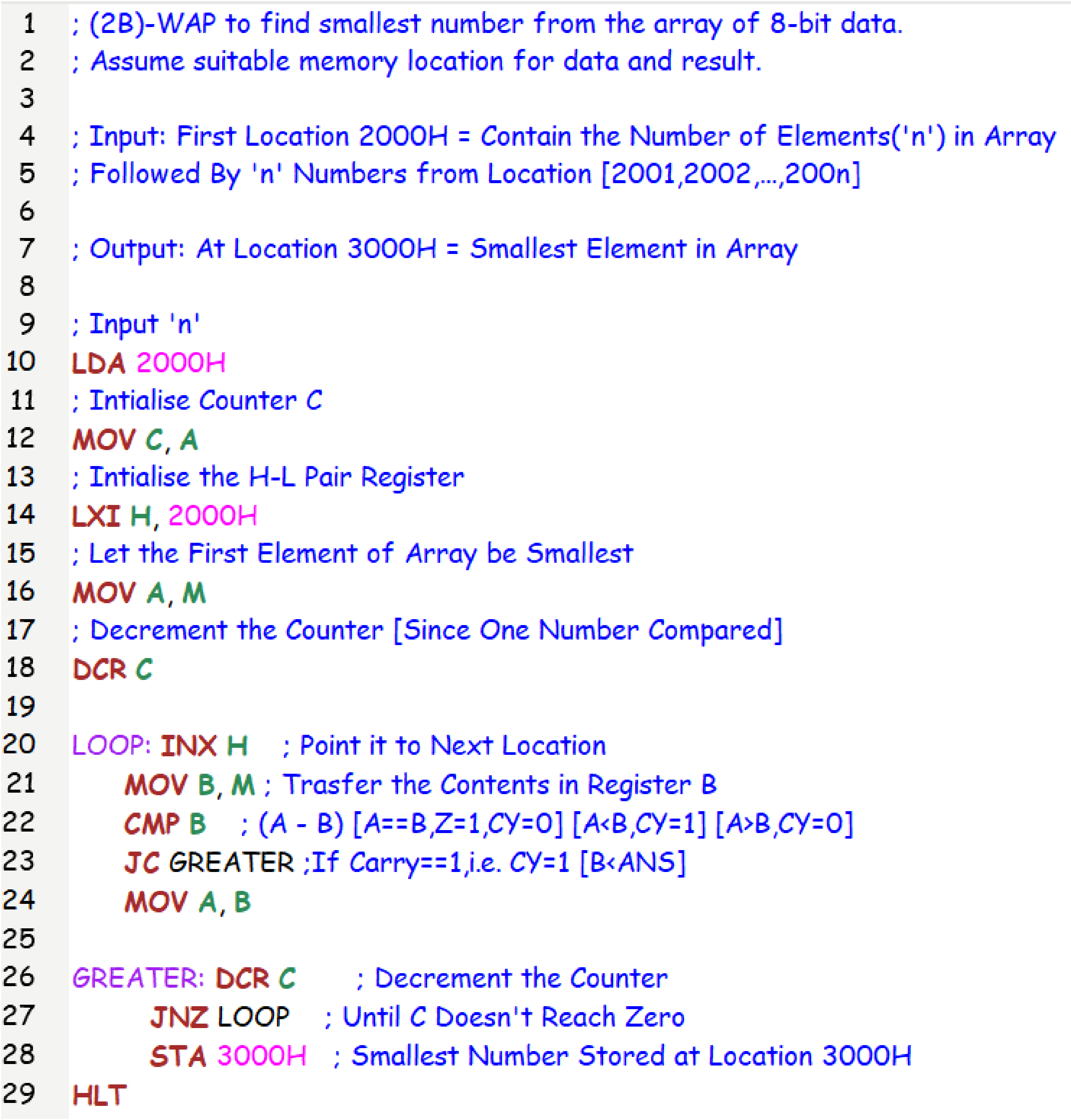


Start 3000h

Address (Hex)	Address	Data
3000	12288	1
3001	12289	0
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully

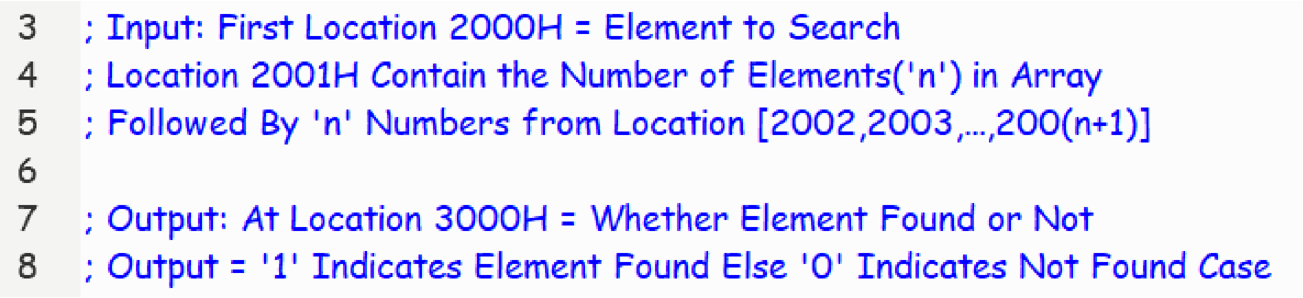
(B) Smallest Element in Array



```
; (2B)-WAP to find smallest number from the array of 8-bit data.
; Assume suitable memory location for data and result.

; Input: First Location 2000H = Contain the Number of Elements('n') in Array
; Followed By 'n' Numbers from Location [2001,2002,...,200n]

; Output: At Location 3000H = Smallest Element in Array

; Input 'n'
LDA 2000H
; Intialise Counter C
MOV C, A
; Intialise the H-L Pair Register
LXI H, 2000H
; Let the First Element of Array be Smallest
MOV A, M
; Decrement the Counter [Since One Number Compared]
DCR C

LOOP: INX H    ; Point it to Next Location
    MOV B, M ; Trasfer the Contents in Register B
    CMP B    ; (A - B) [A==B,Z=1,CY=0] [A<B,CY=1] [A>B,CY=0]
    JC GREATER ;If Carry==1,i.e. CY=1 [B<ANS]
    MOV A, B

GREATER: DCR C      ; Decrement the Counter
      JNZ LOOP    ; Until C Doesn't Reach Zero
      STA 3000H   ; Smallest Number Stored at Location 3000H
HLT
```

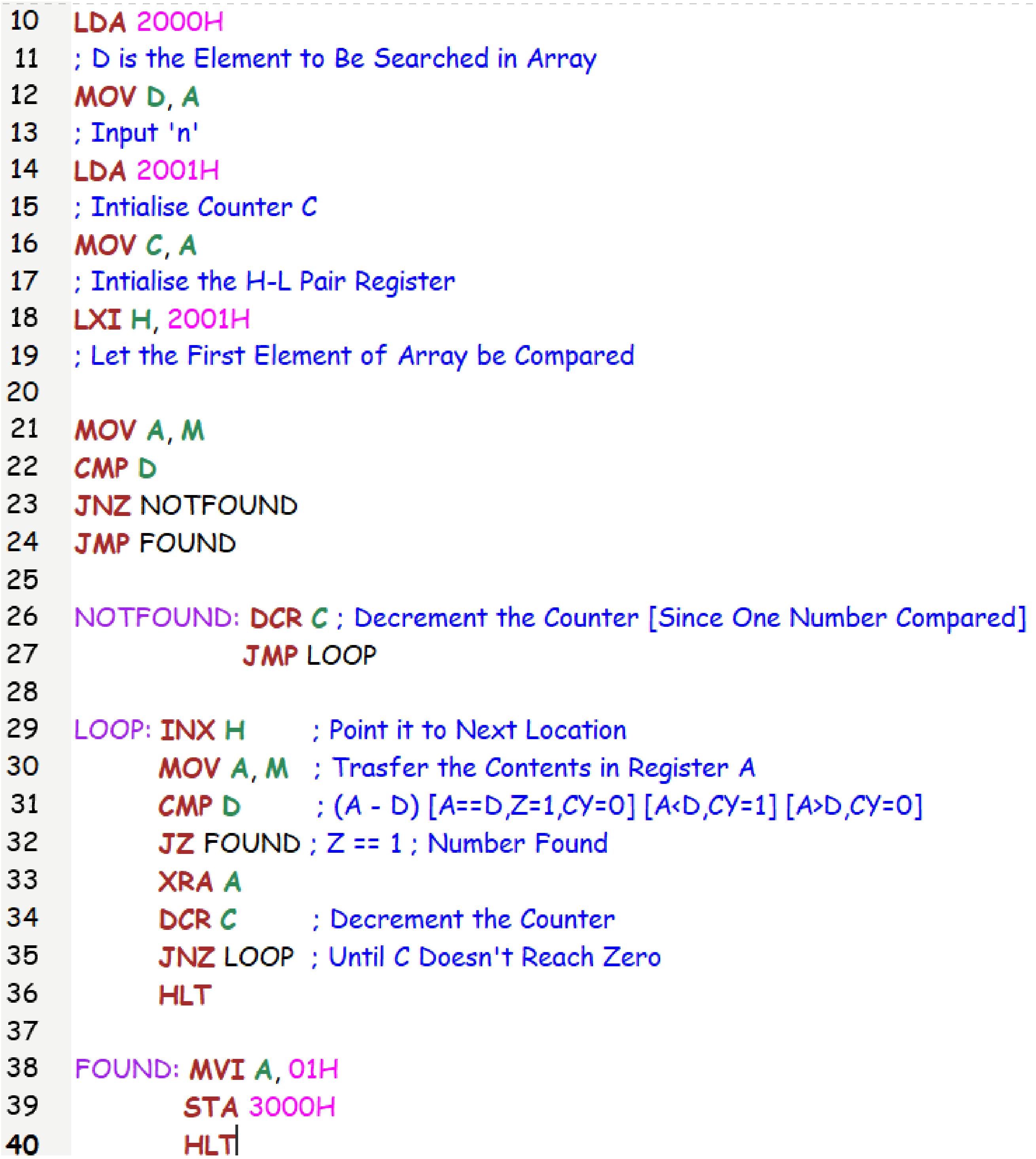
Exampled Attached above Code.

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



```
; Input: First Location 2000H = Element to Search
; Location 2001H Contain the Number of Elements('n') in Array
; Followed By 'n' Numbers from Location [2002,2003,...,200(n+1)]

; Output: At Location 3000H = Whether Element Found or Not
; Output = '1' Indicates Element Found Else '0' Indicates Not Found Case
```

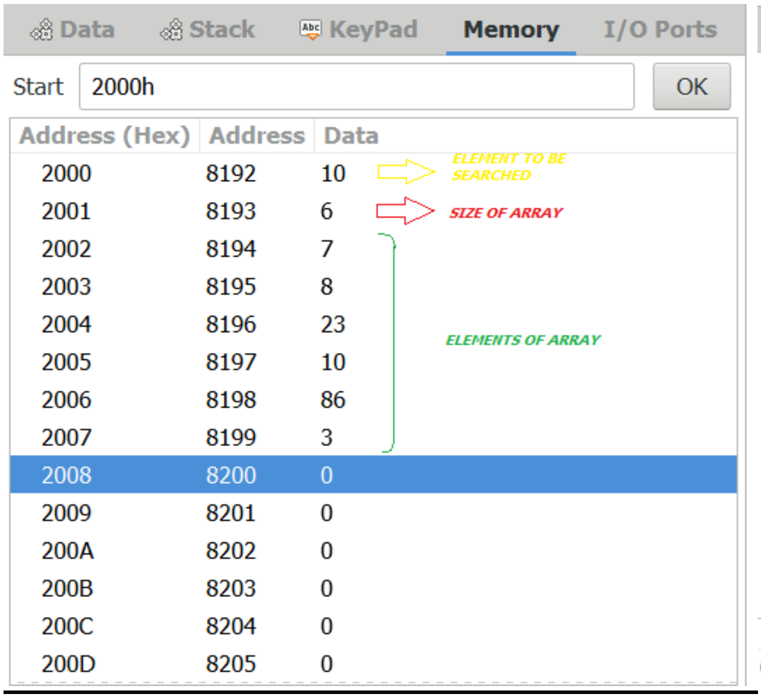
<u>Notepad Code:</u>



```
LDA 2000H
; D is the Element to Be Searched in Array
MOV D, A
; Input 'n'
LDA 2001H
; Intialise Counter C
MOV C, A
; Intialise the H-L Pair Register
LXI H, 2001H
; Let the First Element of Array be Compared

MOV A, M
CMP D
JNZ NOTFOUND
JMP FOUND

NOTFOUND: DCR C ; Decrement the Counter [Since One Number Compared]
          JMP LOOP

LOOP: INX H       ; Point it to Next Location
      MOV A, M    ; Trasfer the Contents in Register A
      CMP D       ; (A - D) [A==D,Z=1,CY=0] [A<D,CY=1] [A>D,CY=0]
      JZ FOUND ; Z == 1 ; Number Found
      XRA A
      DCR C       ; Decrement the Counter
      JNZ LOOP  ; Until C Doesn't Reach Zero
      HLT

FOUND: MVI A, 01H
       STA 3000H
       HLT
```

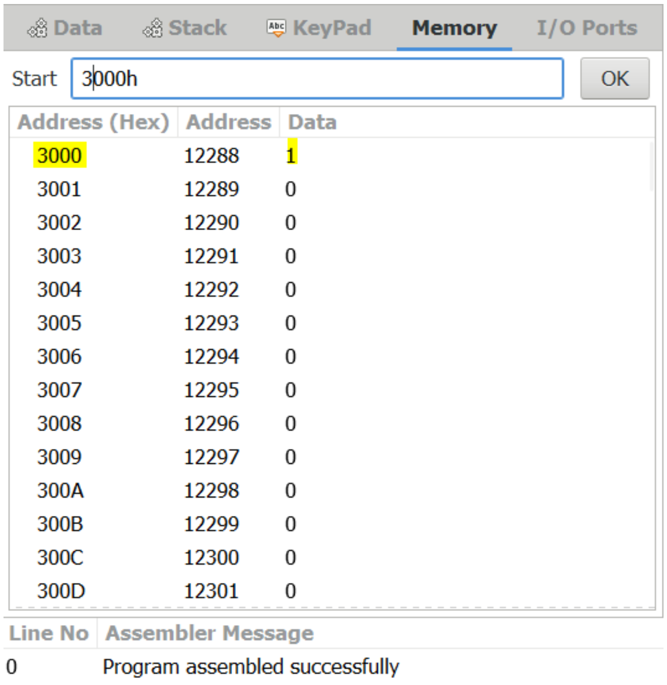
Registers and Memory:

Lets Check for 10 [Element in Array]



Start 2000h

Address (Hex)	Address	Data
2000	8192	10
2001	8193	6
2002	8194	7
2003	8195	8
2004	8196	23
2005	8197	10
2006	8198	86
2007	8199	3
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0
200C	8204	0
200D	8205	0

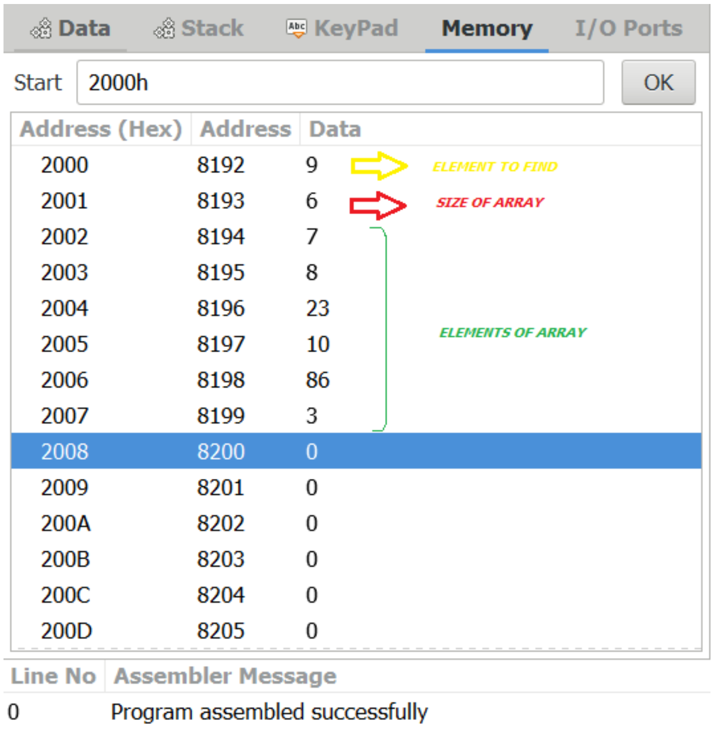


Start 3000h

Address (Hex)	Address	Data
3000	12288	1
3001	12289	0
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully

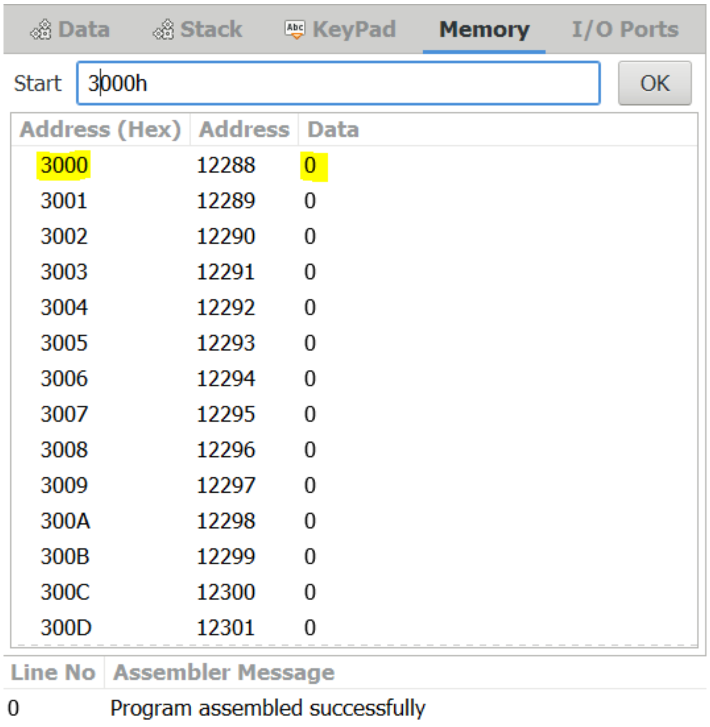
Lets Now Check for 9 [Element Not in Array]



Start 2000h

Address (Hex)	Address	Data
2000	8192	9
2001	8193	6
2002	8194	7
2003	8195	8
2004	8196	23
2005	8197	10
2006	8198	86
2007	8199	3
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0
200C	8204	0
200D	8205	0

Line No	Assembler Message
0	Program assembled successfully



Start 3000h

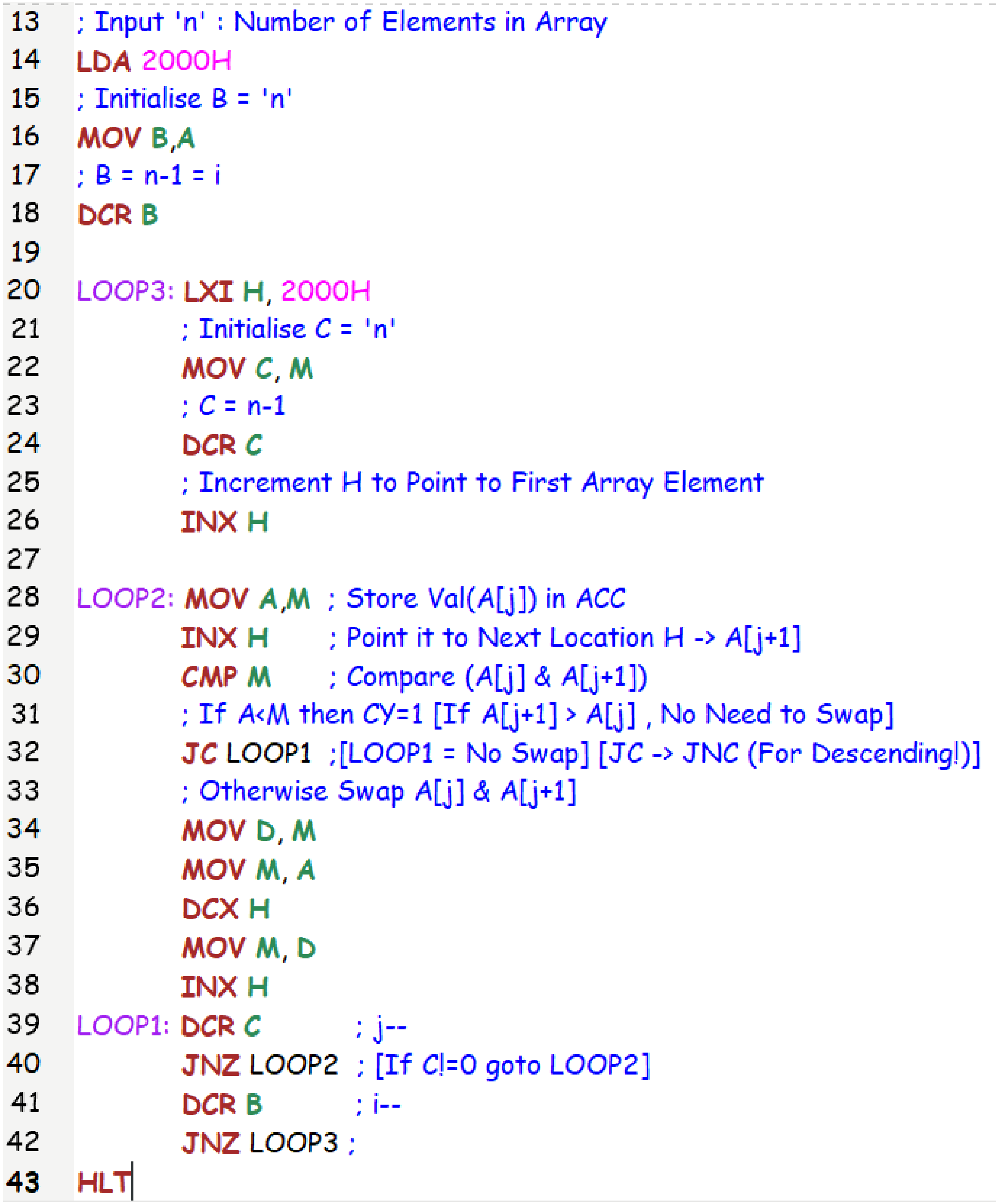
Address (Hex)	Address	Data
3000	12288	0
3001	12289	0
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully

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

```
; Input : First Location 2000H = Contains the Number of Elements ('n') in Array
; Followed by 'n' Numbers From Location [2001, 2002, …, 200n]

; Output : Sorted in the Save Location [2001, 2002, …, 200n]
; Saved Space! Bubble Sort is Inplace Sorting!
```

Notepad Code [Ascending Order]:



```
; Input 'n' : Number of Elements in Array
LDA 2000H
; Initialise B = 'n'
MOV B,A
; B = n-1 = i
DCR B

LOOP3: LXI H, 2000H
       ; Initialise C = 'n'
       MOV C, M
       ; C = n-1
       DCR C
       ; Increment H to Point to First Array Element
       INX H

LOOP2: MOV A,M  ; Store Val(A[j]) in ACC
       INX H    ; Point it to Next Location H -> A[j+1]
       CMP M    ; Compare (A[j] & A[j+1])
       ; If A<M then CY=1 [If A[j+1] > A[j] , No Need to Swap]
       JC LOOP1 ;[LOOP1 = No Swap] [JC -> JNC (For Descending!)]
       ; Otherwise Swap A[j] & A[j+1]
       MOV D, M
       MOV M, A
       DCX H
       MOV M, D
       INX H
LOOP1: DCR C      ; j--
       JNZ LOOP2  ; [If C!=0 goto LOOP2]
       DCR B      ; i--
       JNZ LOOP3 ;
HLT
```

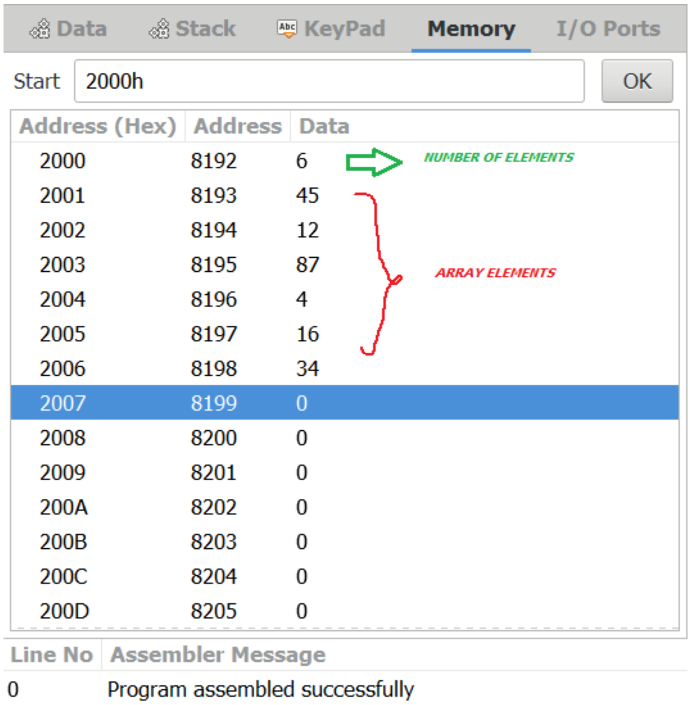
Registers and Memory:

A.) Registers and Memory [Ascending Order]:

Eg:

Input: Array of 6 Numbers = [45, 12, 87, 4, 16, 34]

Output: Array = [4, 12, 16, 34, 45, 87]

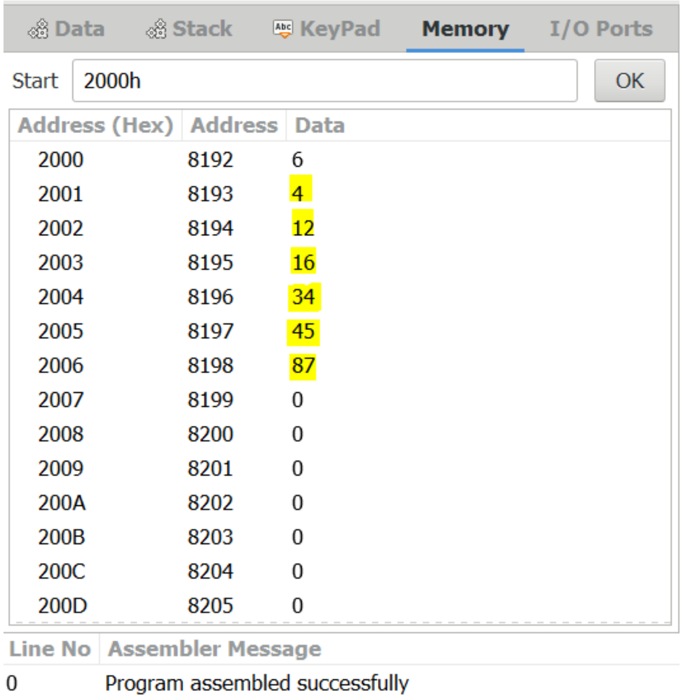


Address (Hex)	Address	Data
2000	8192	6
2001	8193	45
2002	8194	12
2003	8195	87
2004	8196	4
2005	8197	16
2006	8198	34
2007	8199	0
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0
200C	8204	0
200D	8205	0

NUMBER OF ELEMENTS

ARRAY ELEMENTS

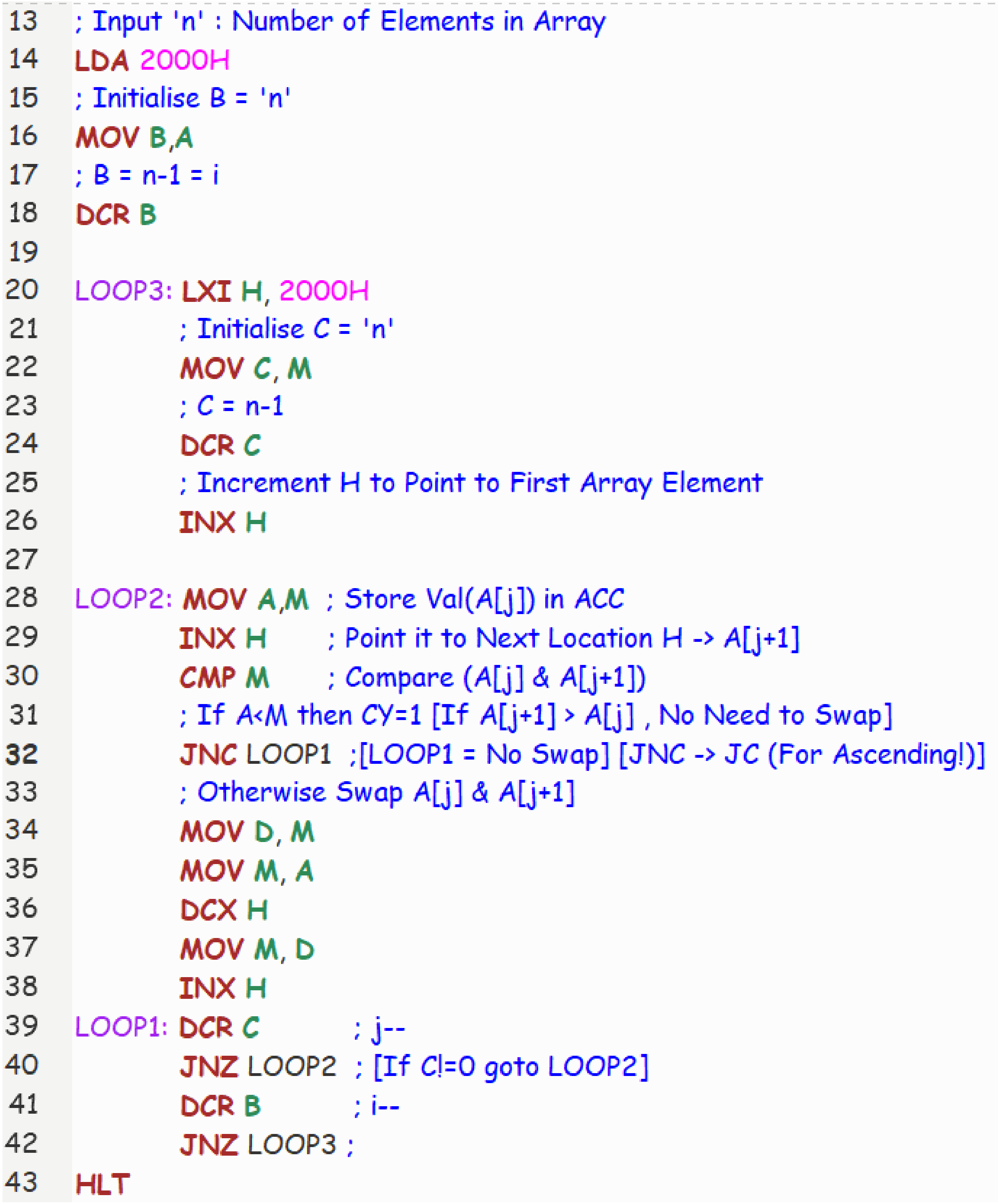
Line No	Assembler Message
0	Program assembled successfully



Address (Hex)	Address	Data
2000	8192	6
2001	8193	4
2002	8194	12
2003	8195	16
2004	8196	34
2005	8197	45
2006	8198	87
2007	8199	0
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0
200C	8204	0
200D	8205	0

Line No	Assembler Message
0	Program assembled successfully

Notepad Code [Descending Order]:



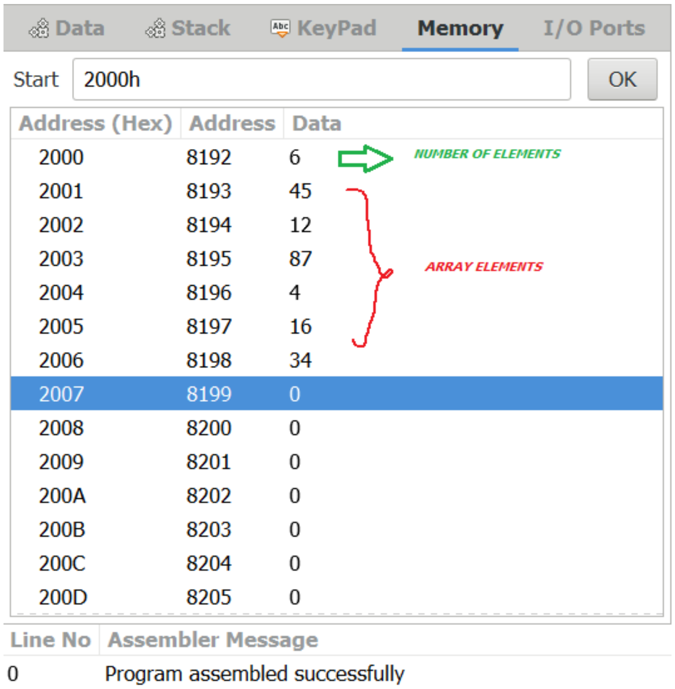
```
; Input 'n' : Number of Elements in Array
LDA 2000H
; Initialise B = 'n'
MOV B,A
; B = n-1 = i
DCR B

LOOP3: LXI H, 2000H
       ; Initialise C = 'n'
       MOV C, M
       ; C = n-1
       DCR C
       ; Increment H to Point to First Array Element
       INX H

LOOP2: MOV A,M  ; Store Val(A[j]) in ACC
       INX H    ; Point it to Next Location H -> A[j+1]
       CMP M    ; Compare (A[j] & A[j+1])
       ; If A<M then CY=1 [If A[j+1] > A[j] , No Need to Swap]
       JNC LOOP1  ;[LOOP1 = No Swap] [JNC -> JC (For Ascending!)]
       ; Otherwise Swap A[j] & A[j+1]
       MOV D, M
       MOV M, A
       DCX H
       MOV M, D
       INX H
LOOP1: DCR C       ; j--
       JNZ LOOP2  ; [If C!=0 goto LOOP2]
       DCR B       ; i--
       JNZ LOOP3 ;
HLT
```

Registers and Memory [Descending Order]:

Eg:

Input: Array of 6 Numbers = [45, 12, 87, 4, 16, 34]

Output: Array = [87, 45, 34, 16, 12, 4]



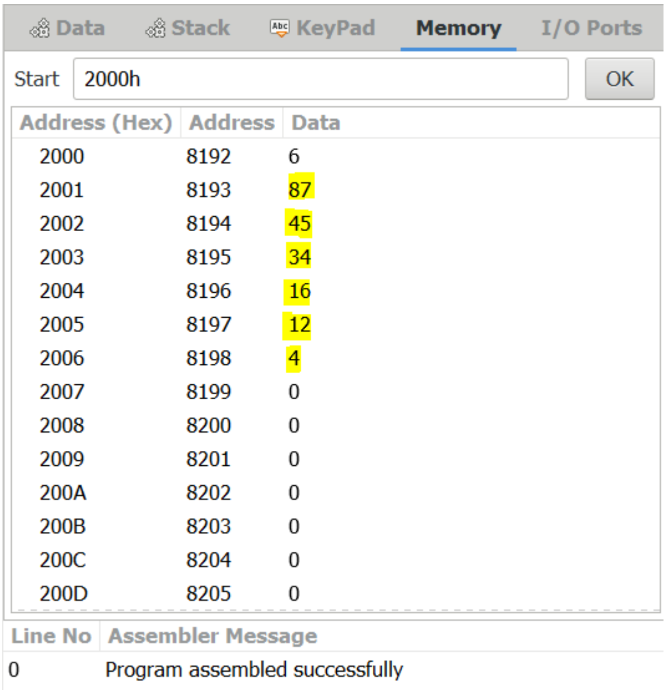
Start 2000h

Address (Hex)	Address	Data
2000	8192	6
2001	8193	45
2002	8194	12
2003	8195	87
2004	8196	4
2005	8197	16
2006	8198	34
2007	8199	0
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0
200C	8204	0
200D	8205	0

NUMBER OF ELEMENTS

ARRAY ELEMENTS

Line No	Assembler Message
0	Program assembled successfully



Start 2000h

Address (Hex)	Address	Data
2000	8192	6
2001	8193	87
2002	8194	45
2003	8195	34
2004	8196	16
2005	8197	12
2006	8198	4
2007	8199	0
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0
200C	8204	0
200D	8205	0

Line No	Assembler Message
0	Program assembled successfully

SUBMITTED BY:

BHAGYA VINOD RANA

[U19CS012]