TITLE: a.       Determine 1’s complement and 2’s complement of any 8-bit number

b.       Determine 1’s complement and 2’s complement of any 16-bit number

PRN: 22510064

DATE: 27/11/2023

AIM: a.       Determine 1’s complement and 2’s complement of any 8-bit number

b.       Determine 1’s complement and 2’s complement of any 16-bit number

HARDWARE USED: Dyna Kit.

SOFTWARE USED: GNU8085 simulator

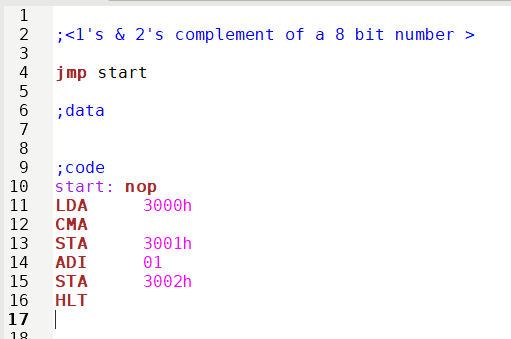
PROCEDURE:

* Install the GNU 8085 simulator software
* Open the simulator and create a new project.
* Write the assembly language code for your program in the editor.
* Assemble the code by clicking on the “assemble” button
* Run the program by clicking on the “run” button.

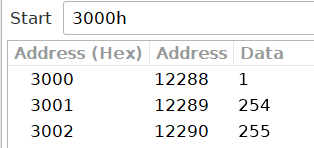
***1’s & 2’s complement of a 8 bit number (GNU SIMULATOR):***

| **Memory** | **Mnemonics** | **Operands** | **Comment** |
| --- | --- | --- | --- |
| 2000 | LDA | [3000] | [A] <- [3000] |
| 2003 | CMA |  | [A] <- [A^] |
| 2004 | STA | [3001] | 1’s complement |
| 2007 | ADI | 01 | [A] <- [A] + 01 |
| 2009 | STA | [3002] | 2’s complement |
| 200C | HLT |  | Stop |

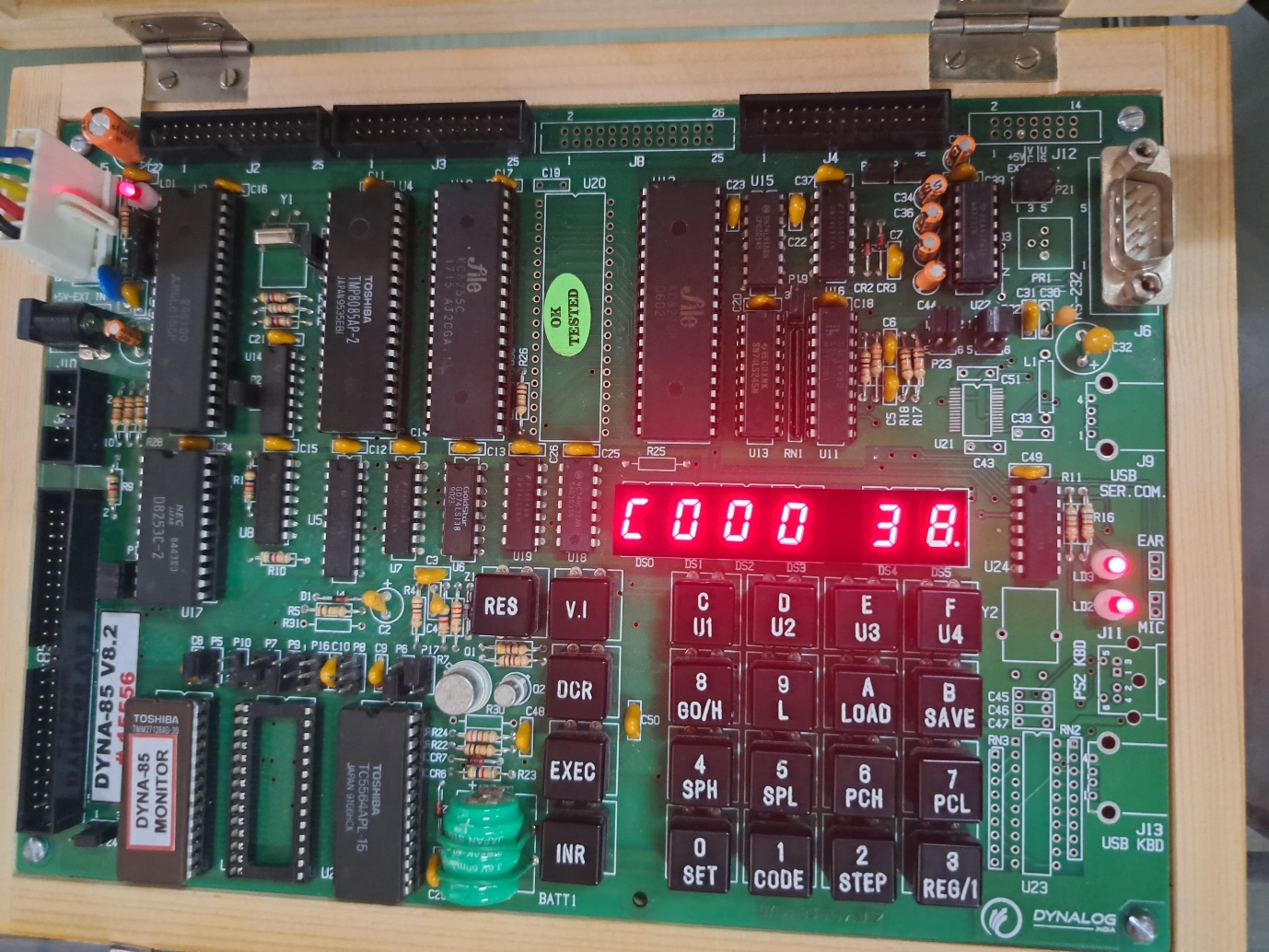
Input:



Output:



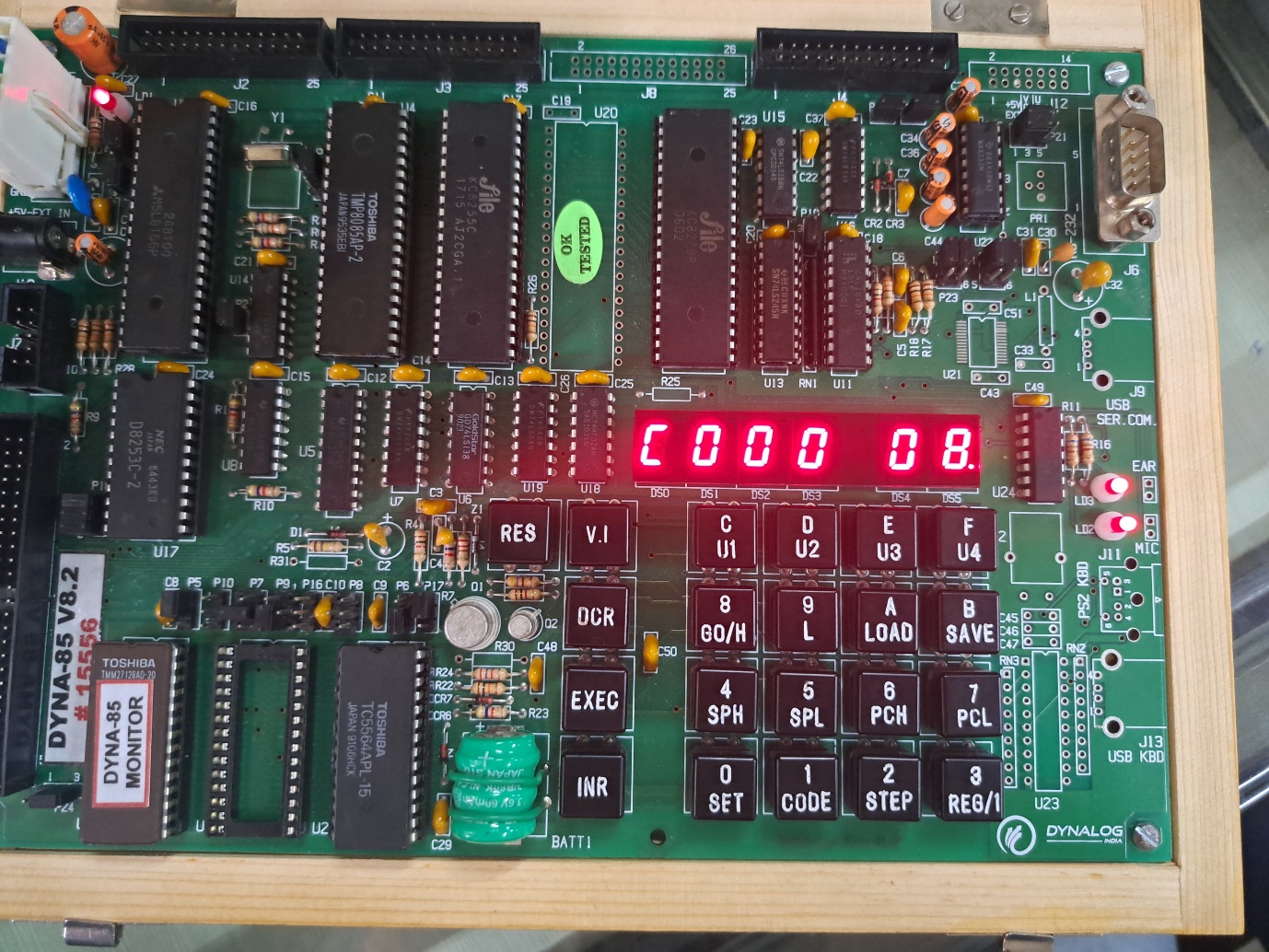
Input for 1’s complement of an 8-bit number:



output for 1’s complement of an 8-bit number:



Input for 2’s complement of an 8-bit number:



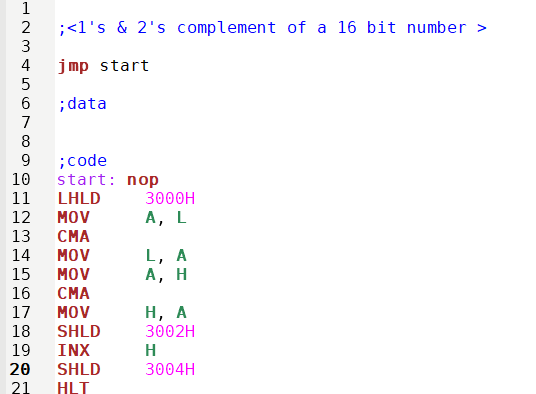
Output for 2’s complement of an 8-bit number:



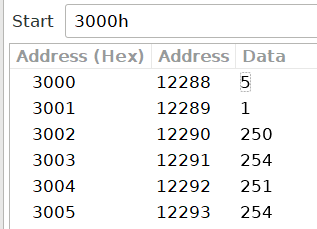
***1’s & 2’s complement of a 16 bit number (GNU SIMULATOR):***

| **Memory** | **Mnemonics** | **Operands** | **Comment** |
| --- | --- | --- | --- |
| 2000 | LHLD | 3000H | [H-L] <- [3000] |
| 2003 | MOV | A, L | [A] <- [L] |
| 2004 | CMA |  | [A] <- [A^] |
| 2005 | MOV | L, A | [L] <- [A] |
| 2006 | MOV | A, H | [A] <- [H] |
| 2007 | CMA |  | [A] <- [A^] |
| 2008 | MOV | H, A | [H] <- [A] |
| 2009 | SHLD | 3002H | 1’s complement |
| 200C | INX | H | [H-L] <- [H-L] + 1 |
| 200D | SHLD | 3003H | 2’s complement |
| 2010 | HLT |  | Stop |

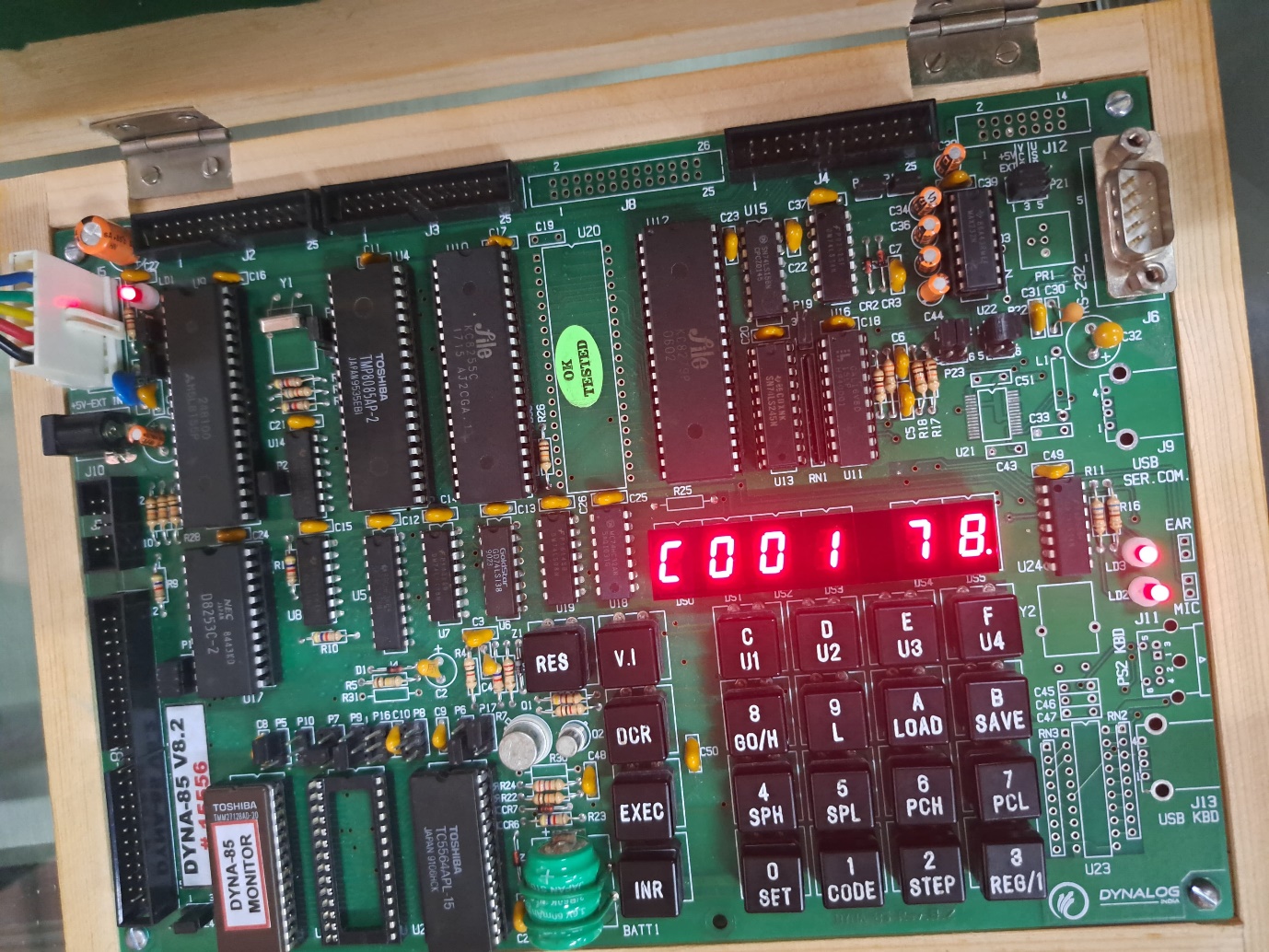
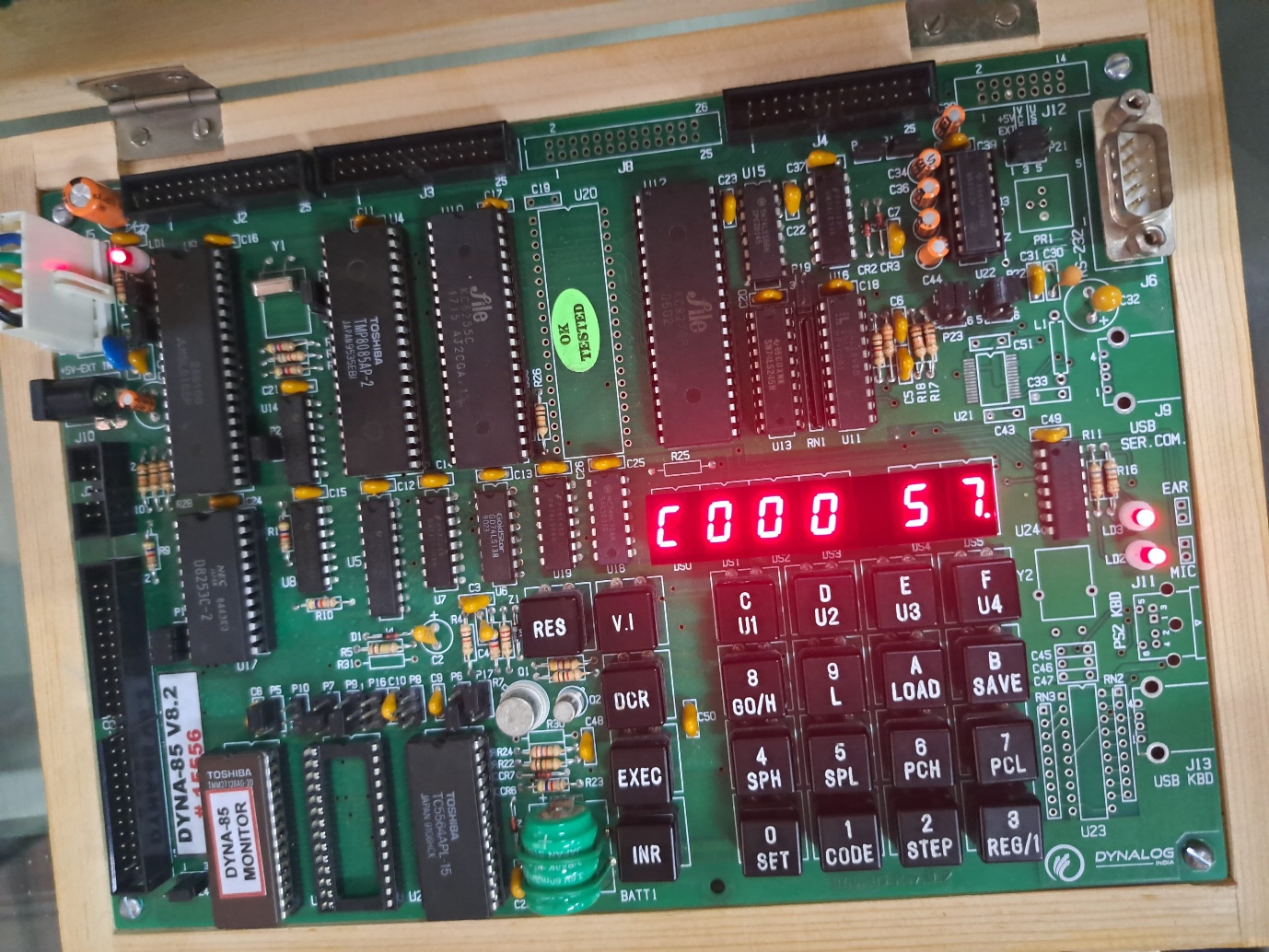
Input:

******

Output:

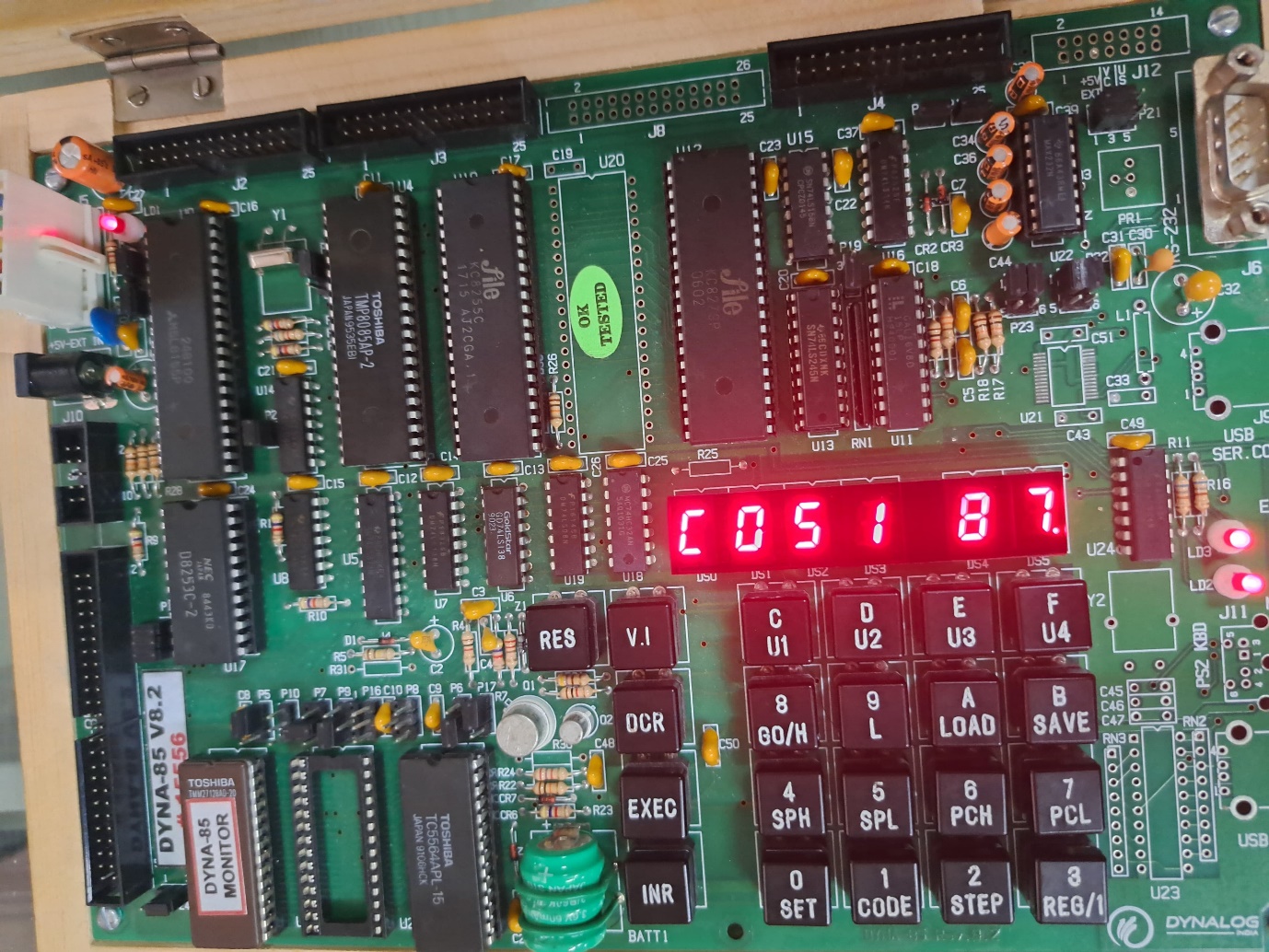
******

Input for 1’s complement of a 16-bit number:

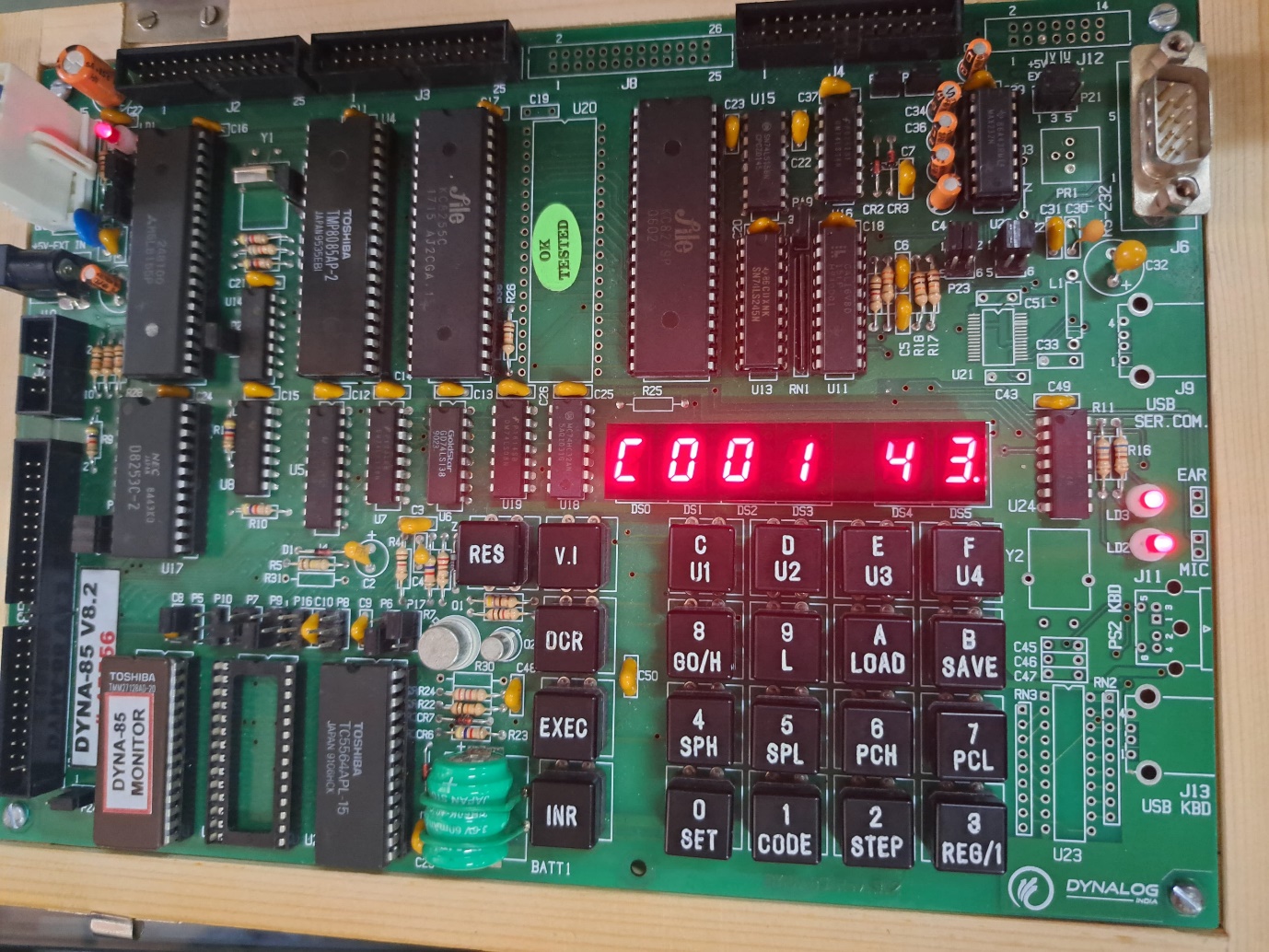
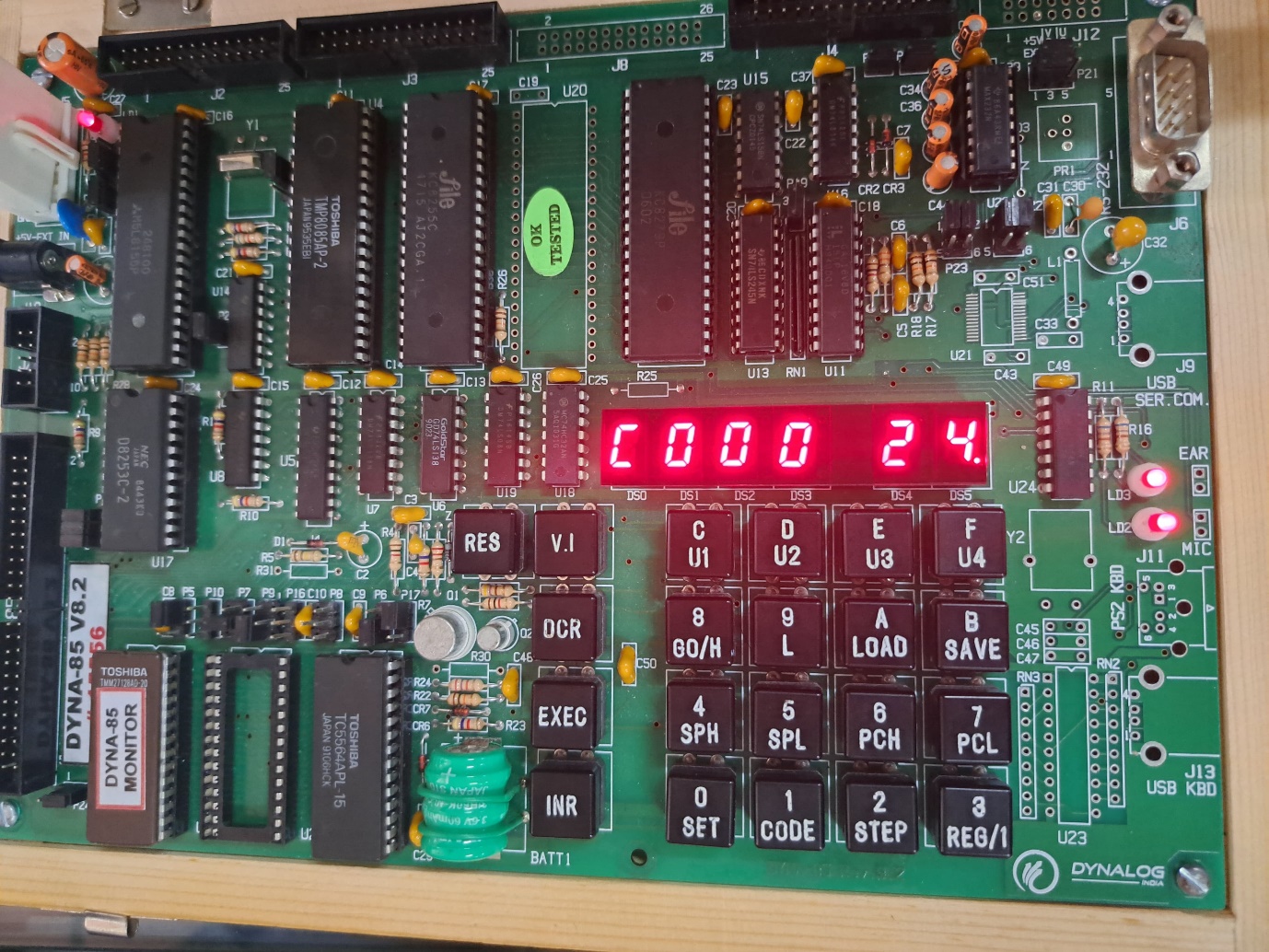


Output for 1’s complement of a 16-bit number:





Input for 2’s complement of a 16-bit number:



Output for 2’s complement of a 16-bit number:

