PRACTICAL 2

**2021**

USIU

Group 6

3/2/2021



**United States International University Africa  
Nairobi, Kenya  
APT2022-Introduction To Assembly Programming**

**PRACTICAL LAB 2: Emulator(EMU8086)**

**LECTURER: LINUS ALOO**

Group 6

* Clifford Ombiro 660904
* Timothy Diero 661405
* Moses Saruni 661821
* Aamir Abdirahman 661392
* Bill Musani 660884
* Charlton Charles 654039
* Alex Rugara 657317
* David Nelson 656429

**PRACTICAL LAB 2:**

**emulator(EMU8086) and 8086Microprocessor**

**Objectives**

* The main idea in this lab exercise is to understand or know what an emulator in this case EMU8086 is and how it is related to the microprocessor8086
* We will also be using the EMU8086 to run some programs and see how they run and by this we will be understanding coding with an emulator,and by so we will be making mathematical calculations such as (addition,subtraction ,e.tc)
* We will understand how codes are sequentially executed by method called single stepping
* By doing this we will be understanding also some features of the emulator such as
* The source editor
* Application of mnemonics
* Converter
* In this lab exercise we will also understand what signed & unsigned numbers are and also how do we put into application the various registers we learned in class and how they store data in the memory and how we can read dater from the registers
* And finally we will understand how the emulator(EMU8086) outputs codes and how we can understand the outputs ,this is by reading through the source file (machine code language)

**Introduction**

In lab 2 exercise we are going to learn emulator 8086 environment and understand how we can assemble instructions to the memory microprocessor 8086 by using EMU8086.this will only be achievable if we write a complete assembly program .

The source editor in the emu8086 is a special editor that understand mnemonics,hexadecimal numbers and it actually labels them by different colors as will be seen in the codes

In this lab we will be developing programs to perform the following arithimetic operations

* (3010+ 1510) \* ( 57510–22510) + 210
* 14 10 \* ( 23310 – 125 10 ) + 227

In this lab we will also be writing and running some codes in emu8086 that would output character message such as a program that displays **“hellow world”**we will get to see that each and every character lets say “h” ,”e” and the rest actually occupies memory

Codes will run by pressing the emulation button and a pop up window will appear we can choose to run the code or we can slowly see the execution of instructions by pressing the “single step” button.

**Material under usage**

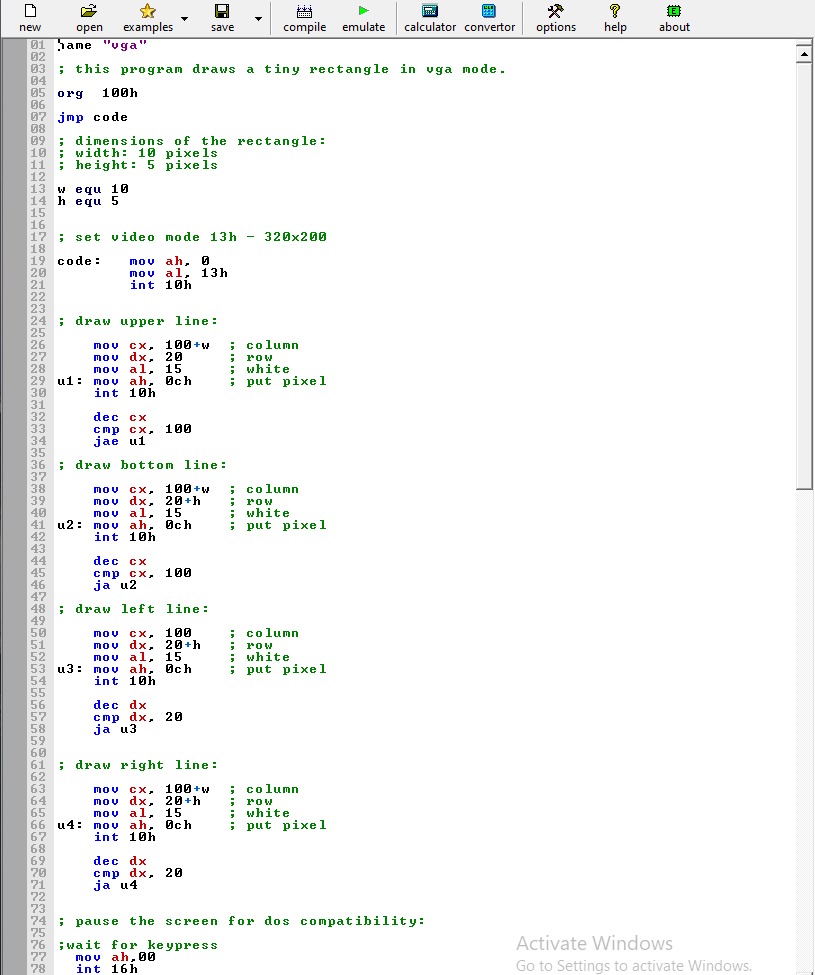
We will be using the following tools in our lab exercise

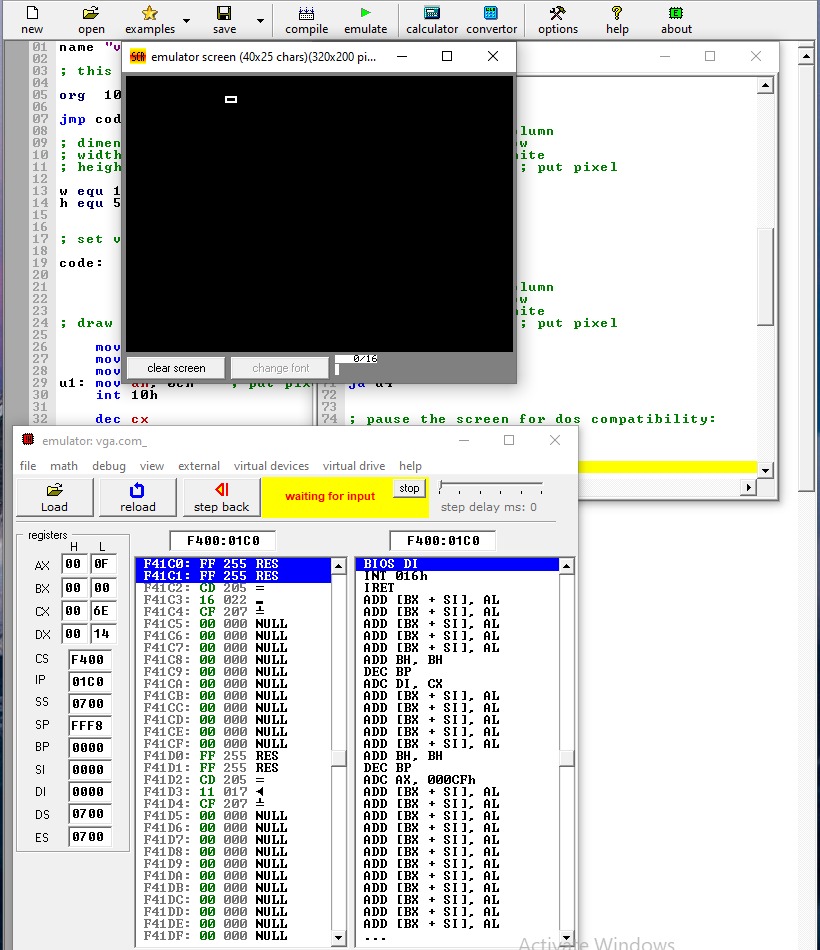
* A computer
* Emulator(EMU8086)
* Internet connection
* Microsoft word

Methodology

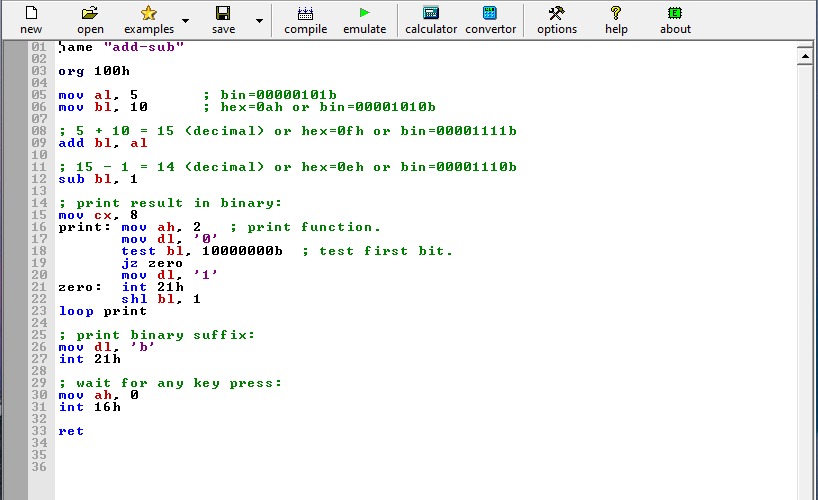
**Part 2: Opt-in Examples in Emu8086**

**VGA**

**[](blob:https://web.whatsapp.com/f41b1728-780e-493c-81c6-34562963301a)**

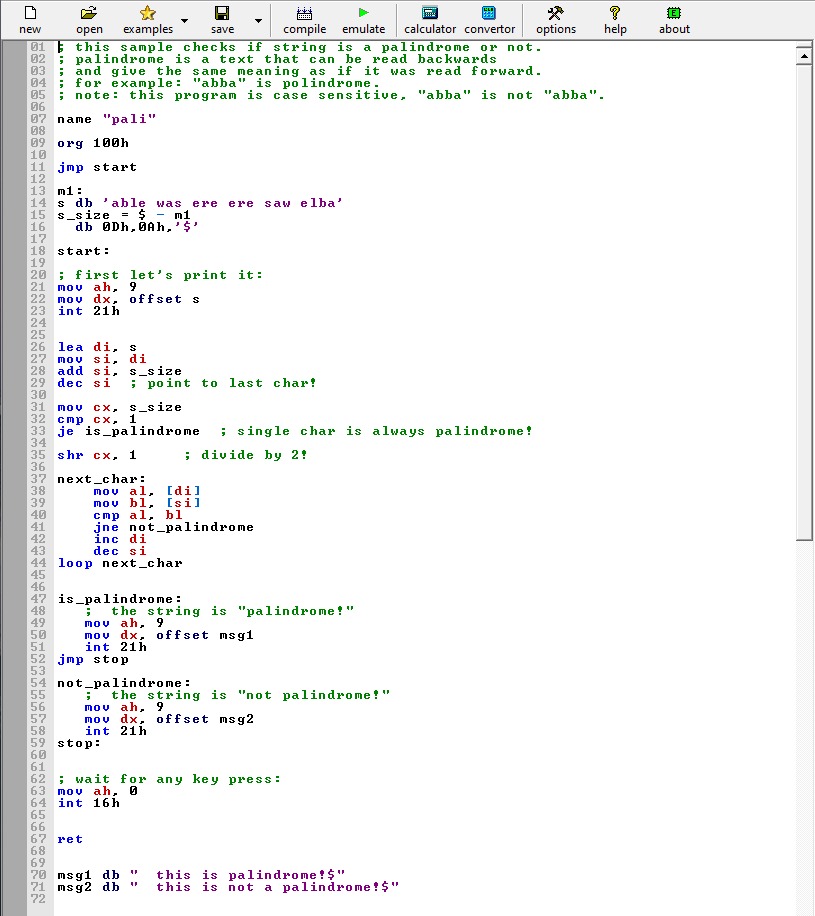


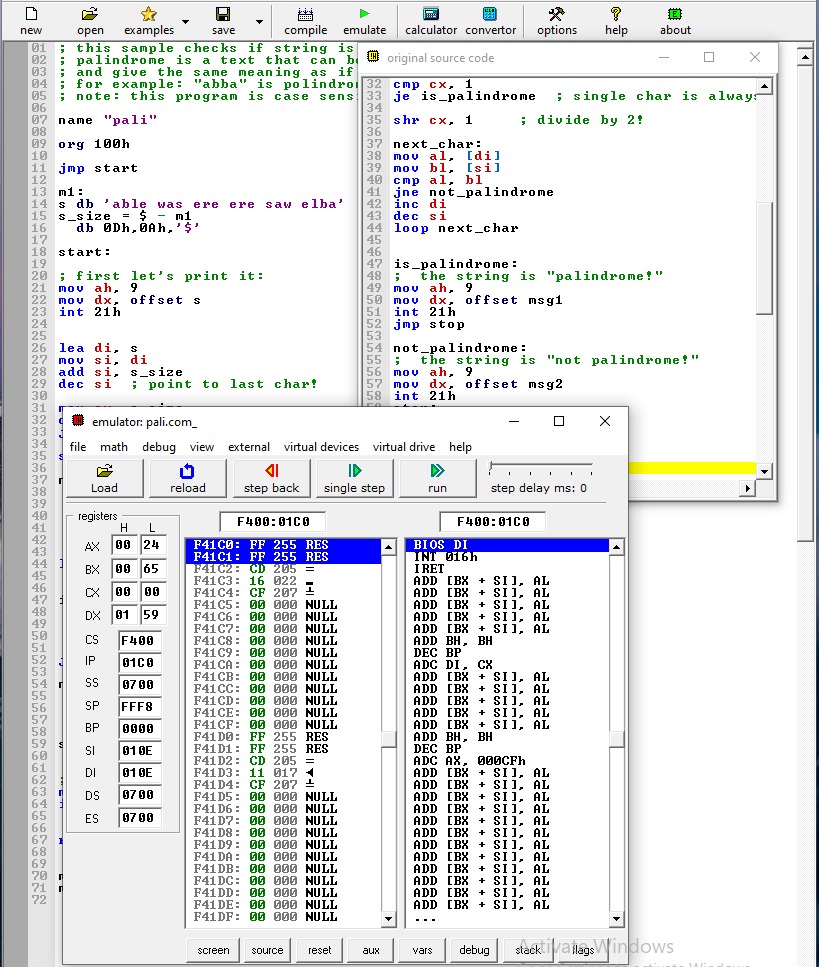
ADD/SUBTRACT



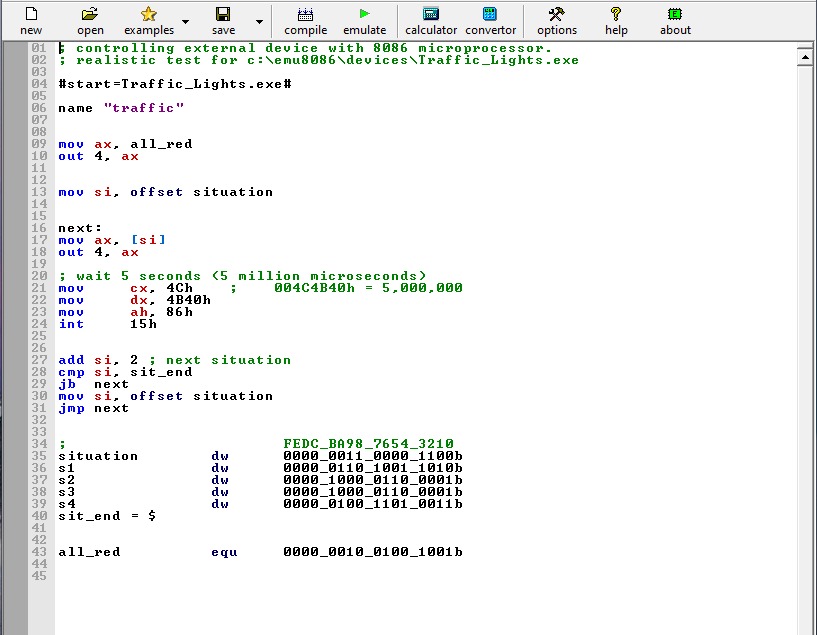


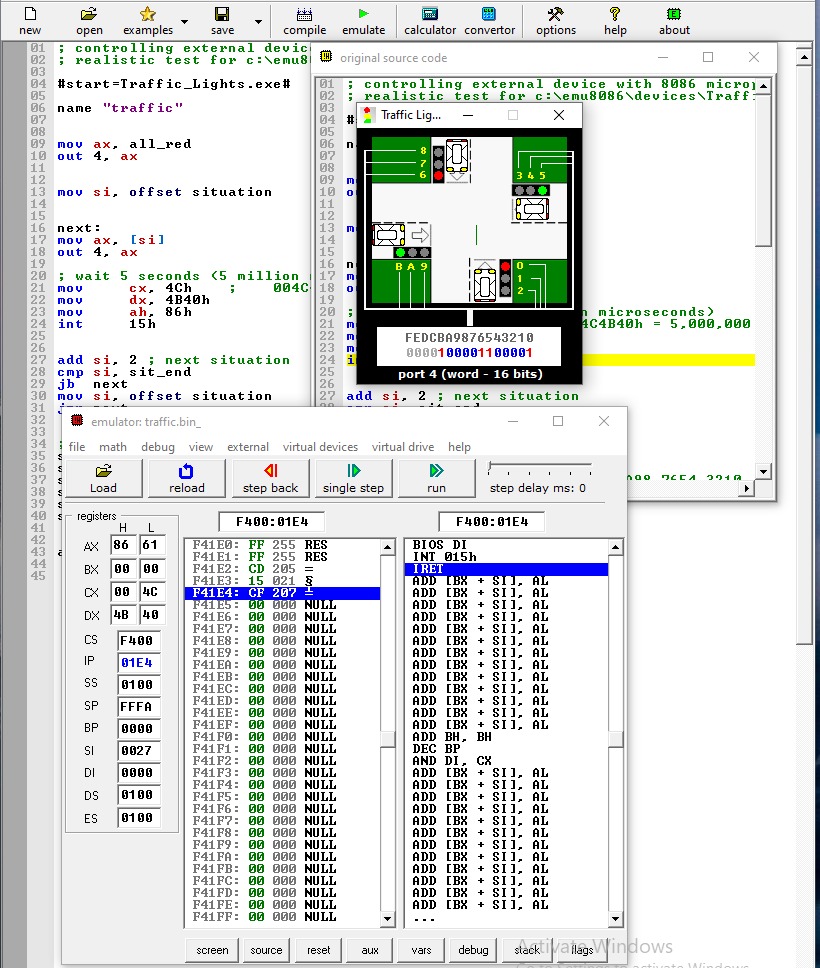
Palindrome



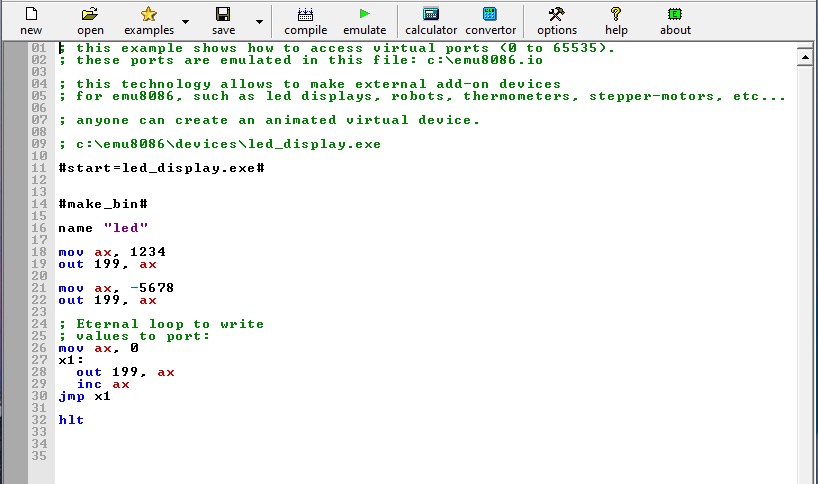


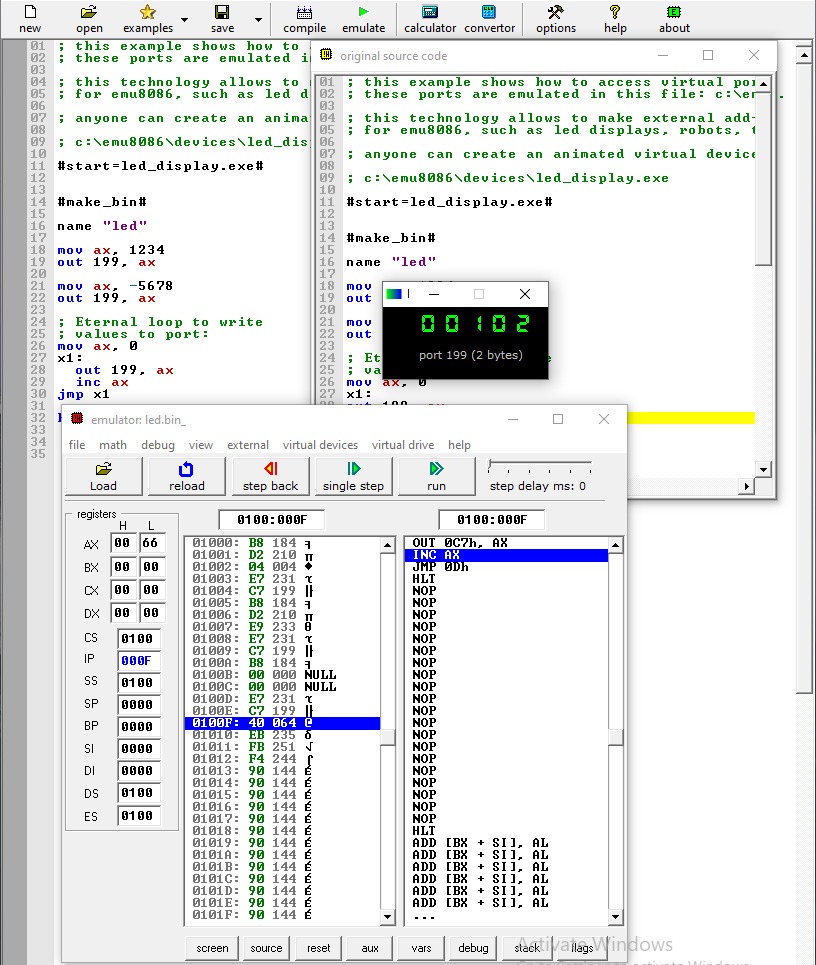
TRAFFIC LIGHTS



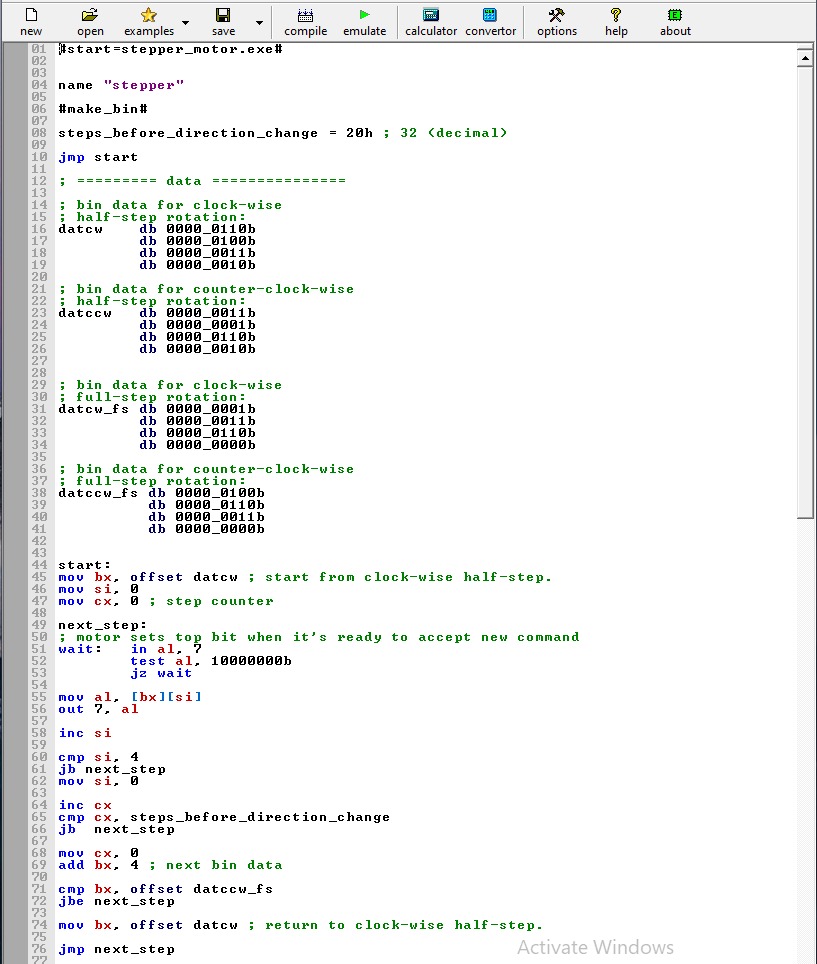


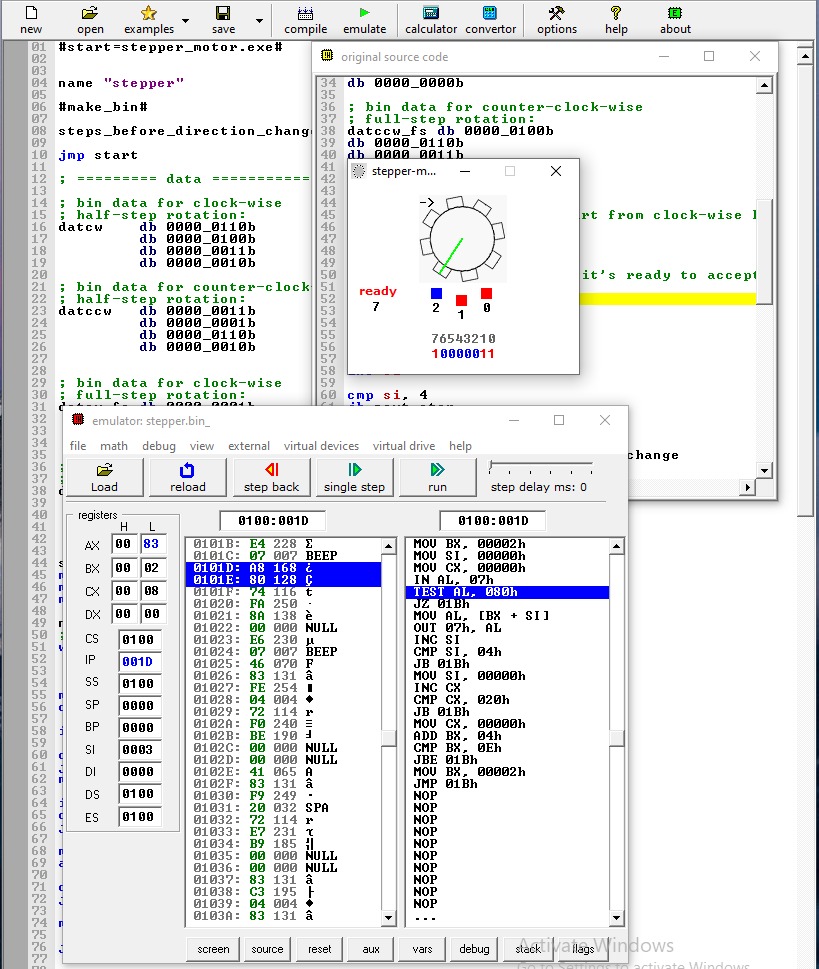
LED DISPLAY



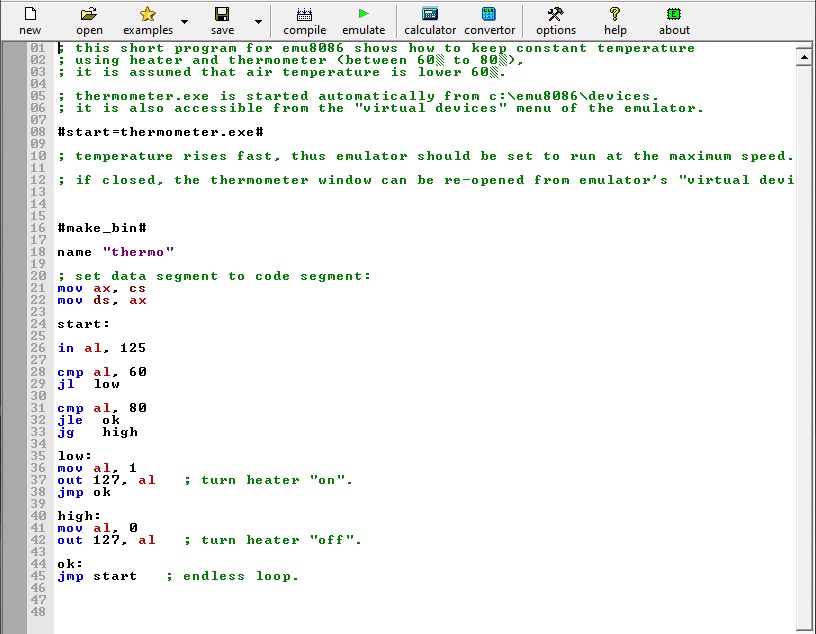


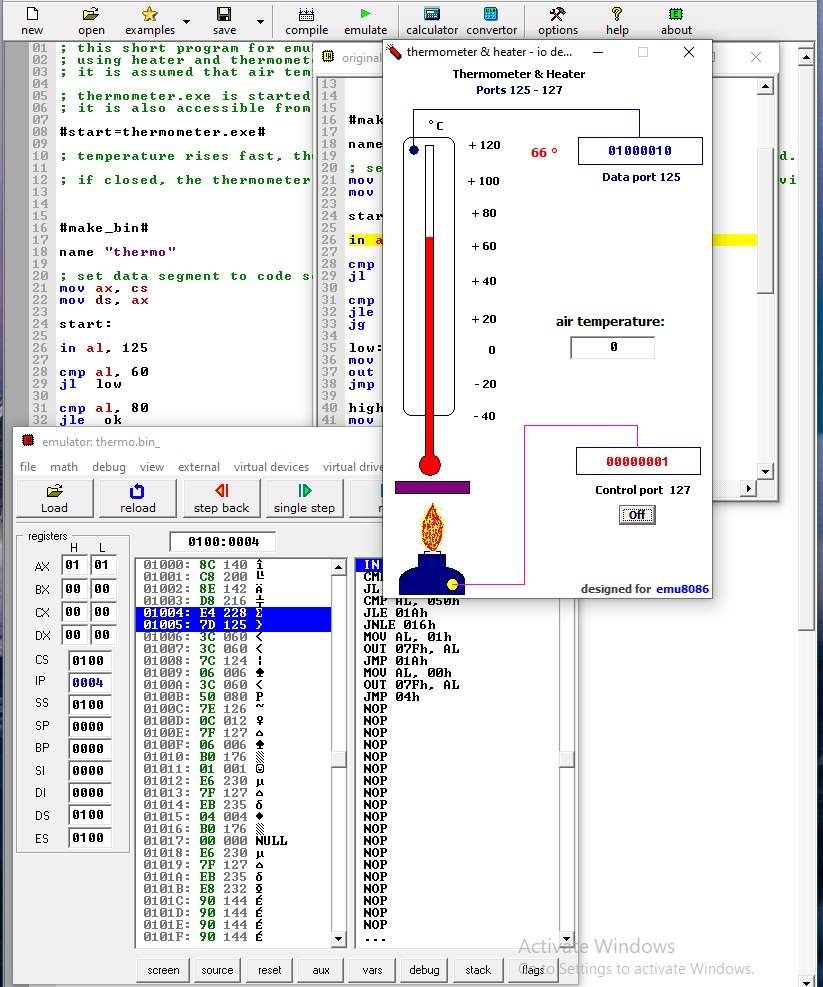
STEPPER MOTOR





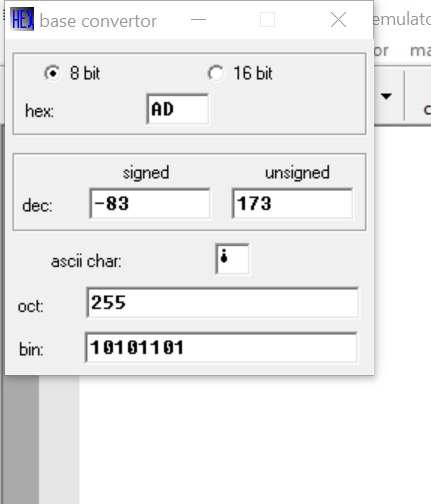
Thermometer



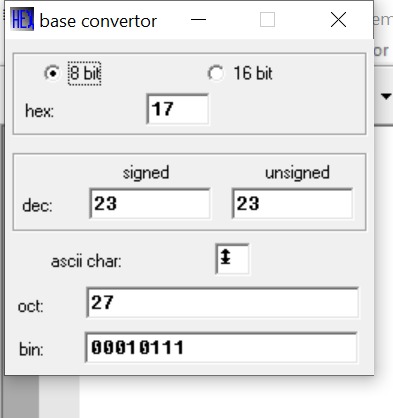


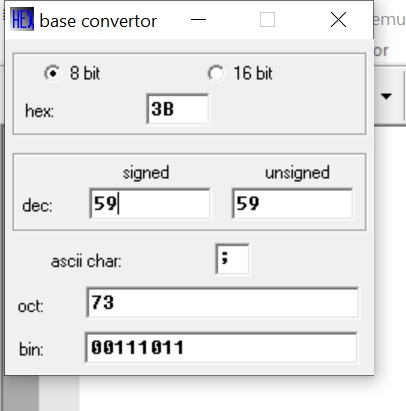
**Part 3: Assemble and execute instructions in Emu8086**

**10101101b**

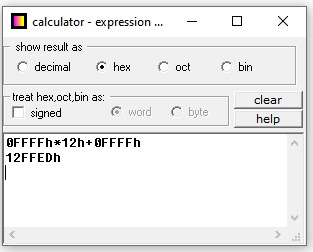


17H

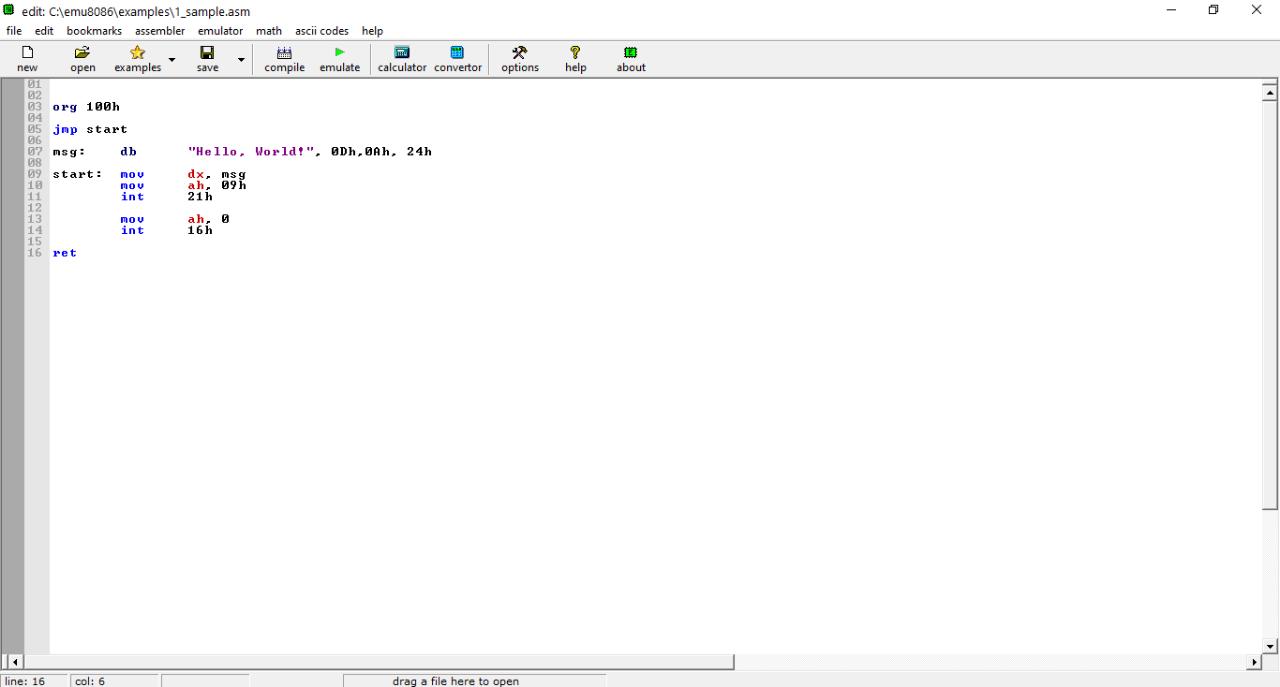


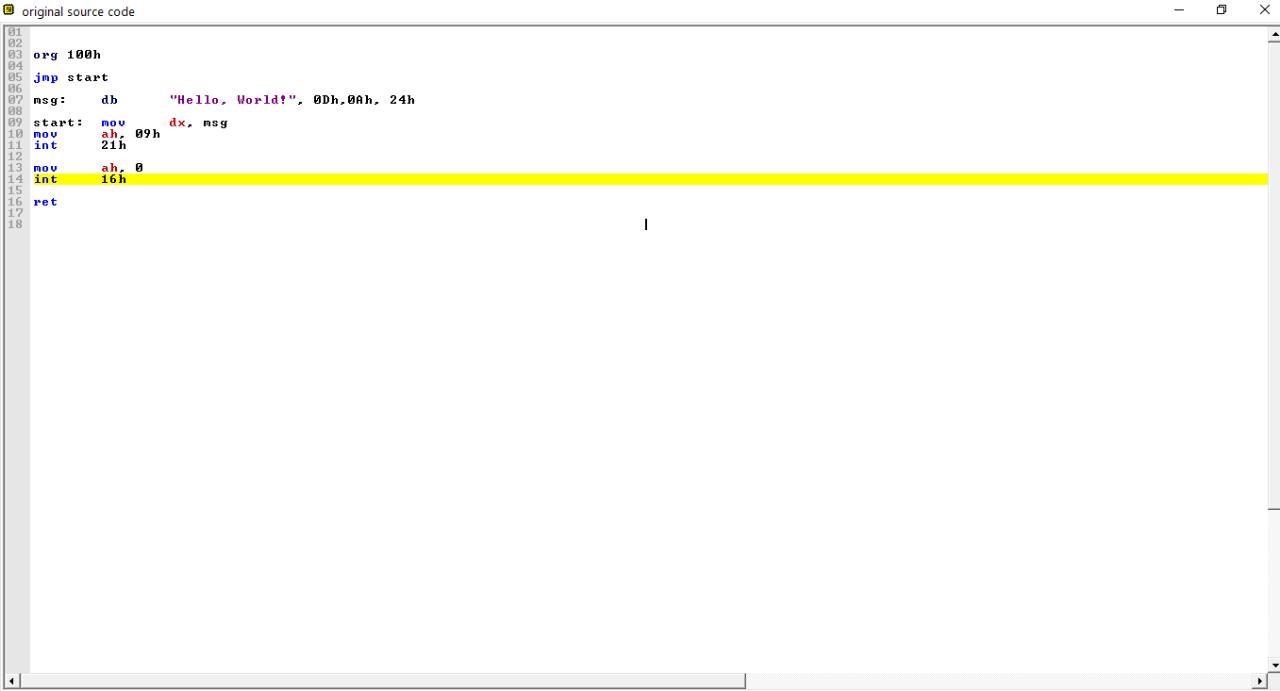
59

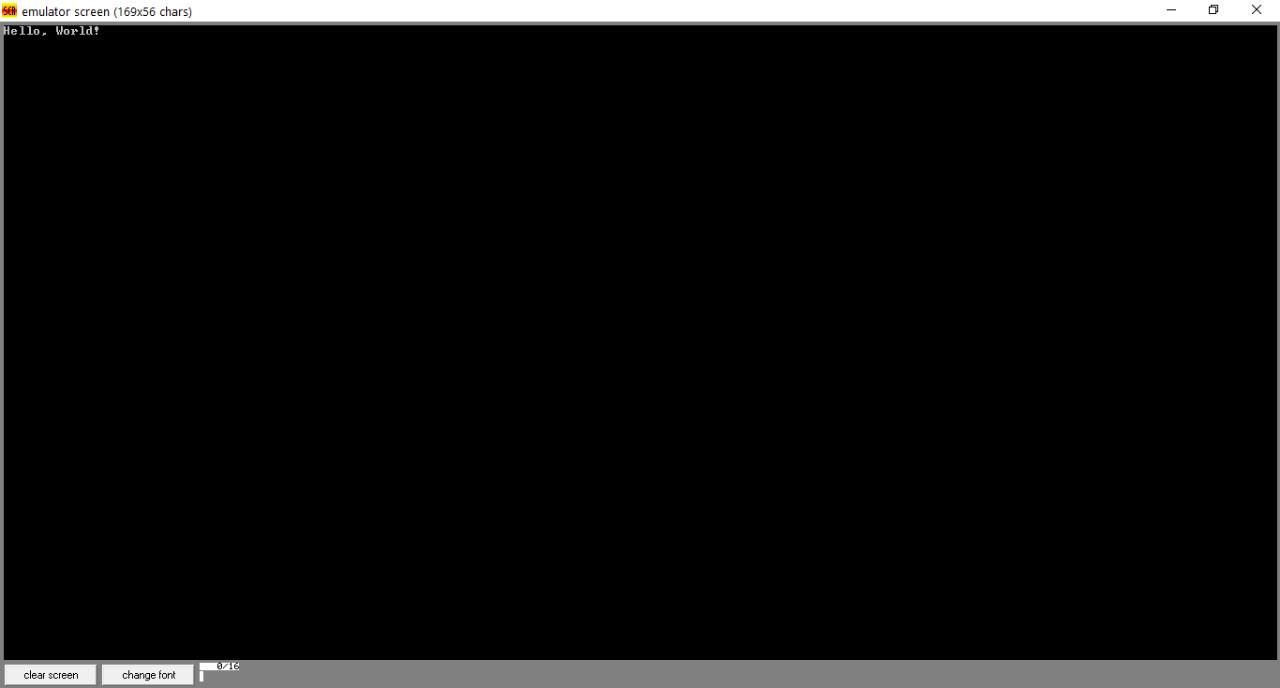
**Use EMU8086 to evaluate an expressions**

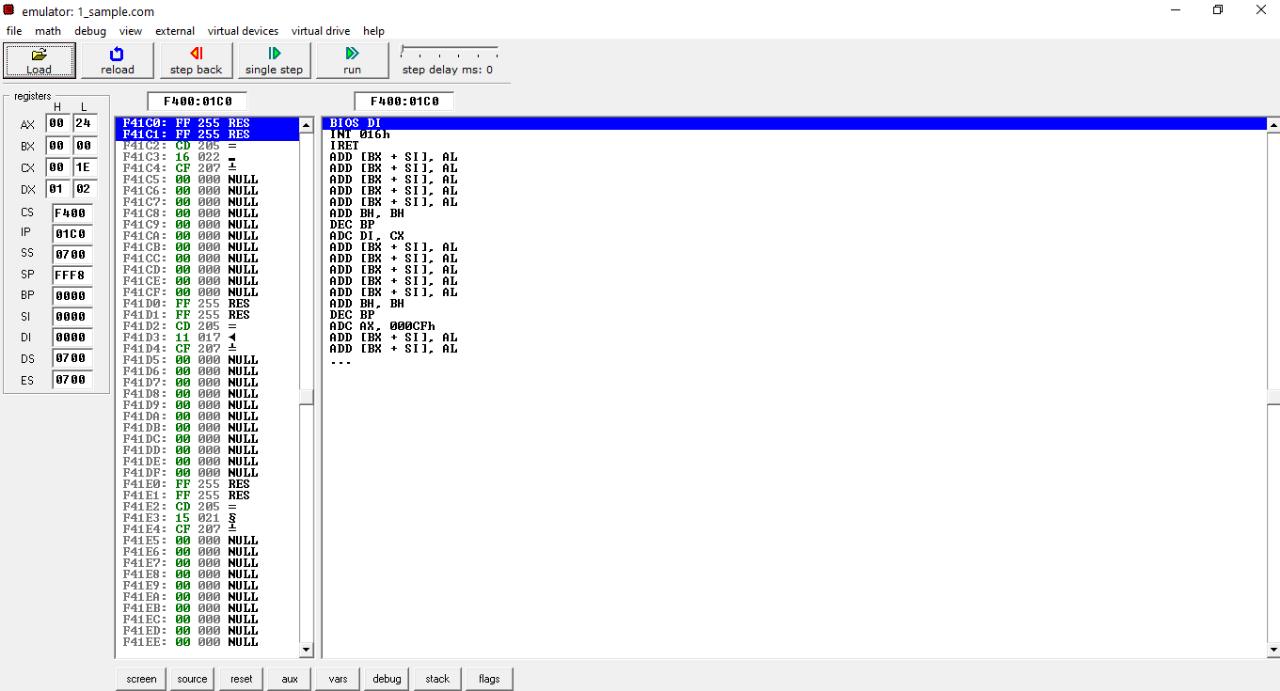
****

**Part 4: Writing and Running Assembly Code in Emu8086**

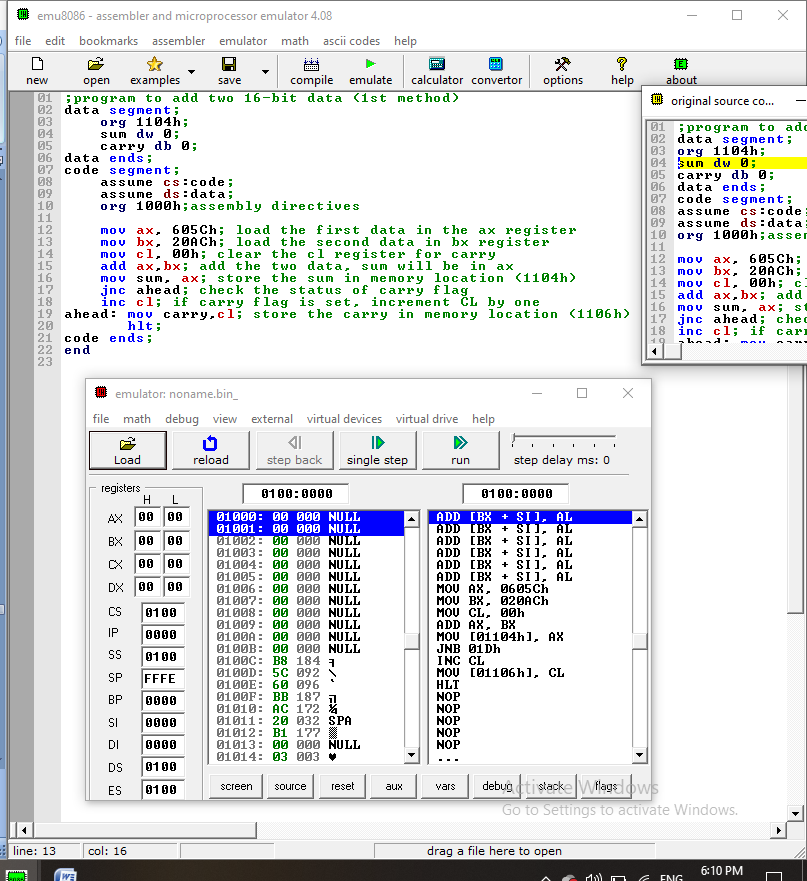
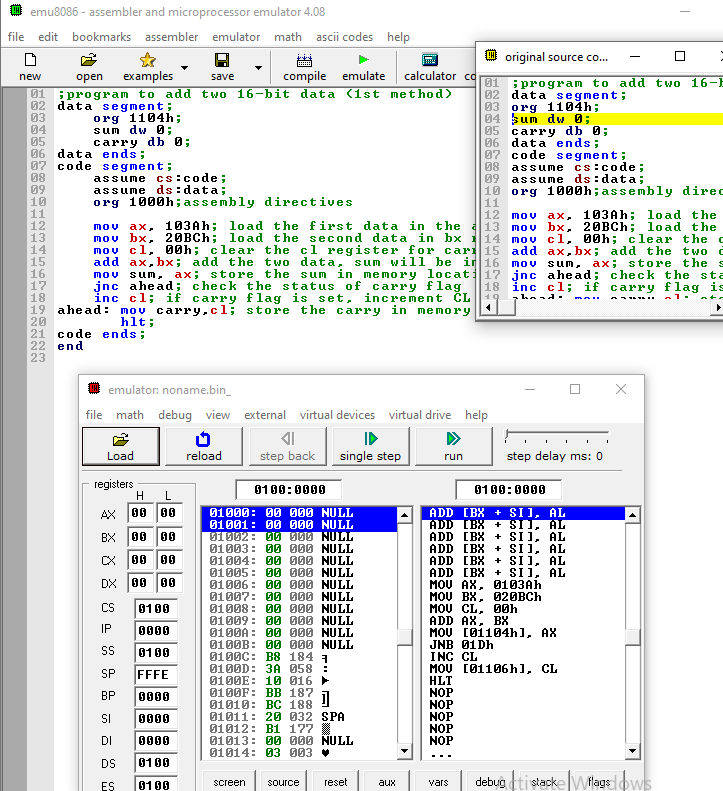
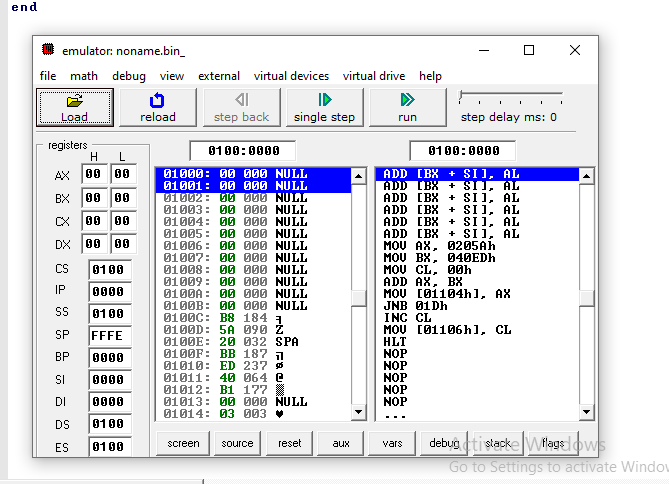
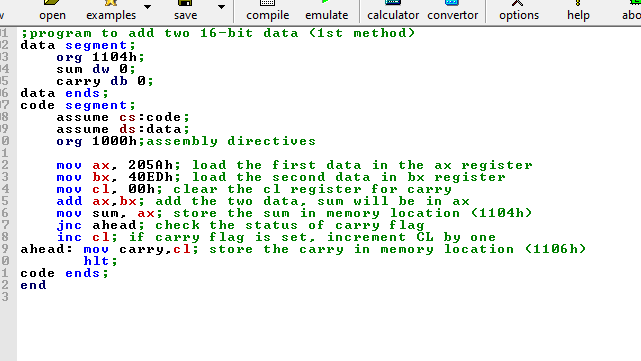
****

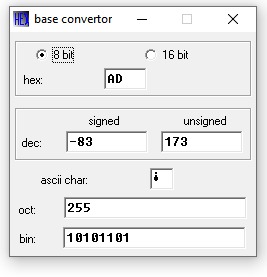






**Part 4: Writing and Running Assembly Code in Emu8086**



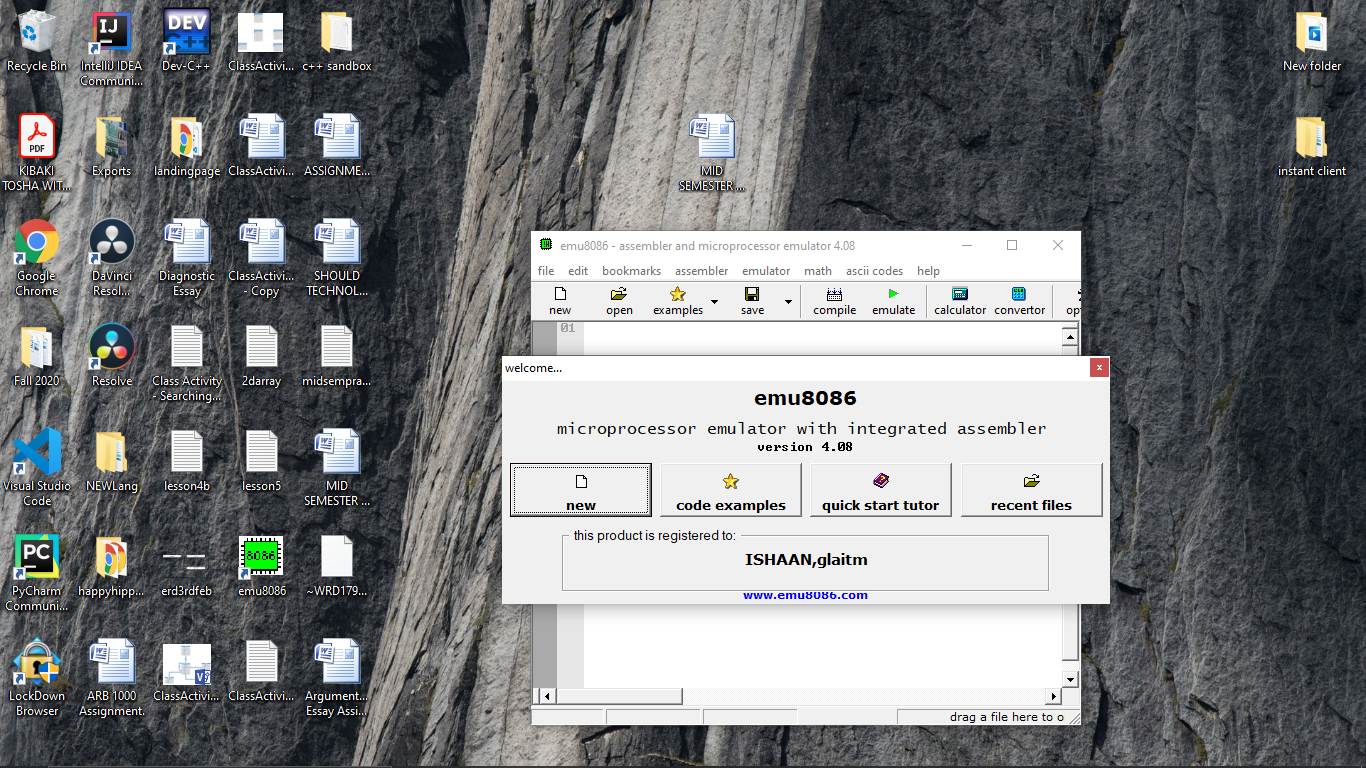


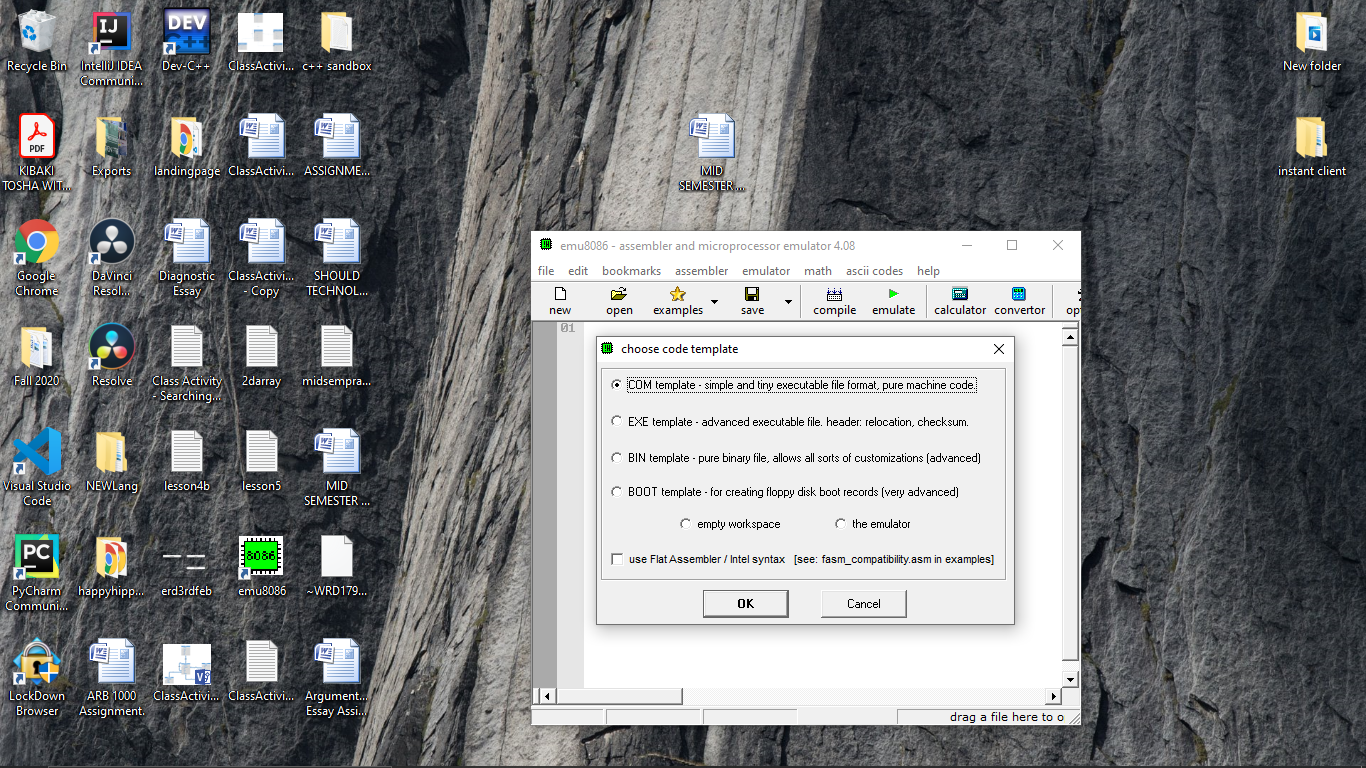
RESULTS AND DISCUSSIONS

**PART 5 : EXERCISE PART**

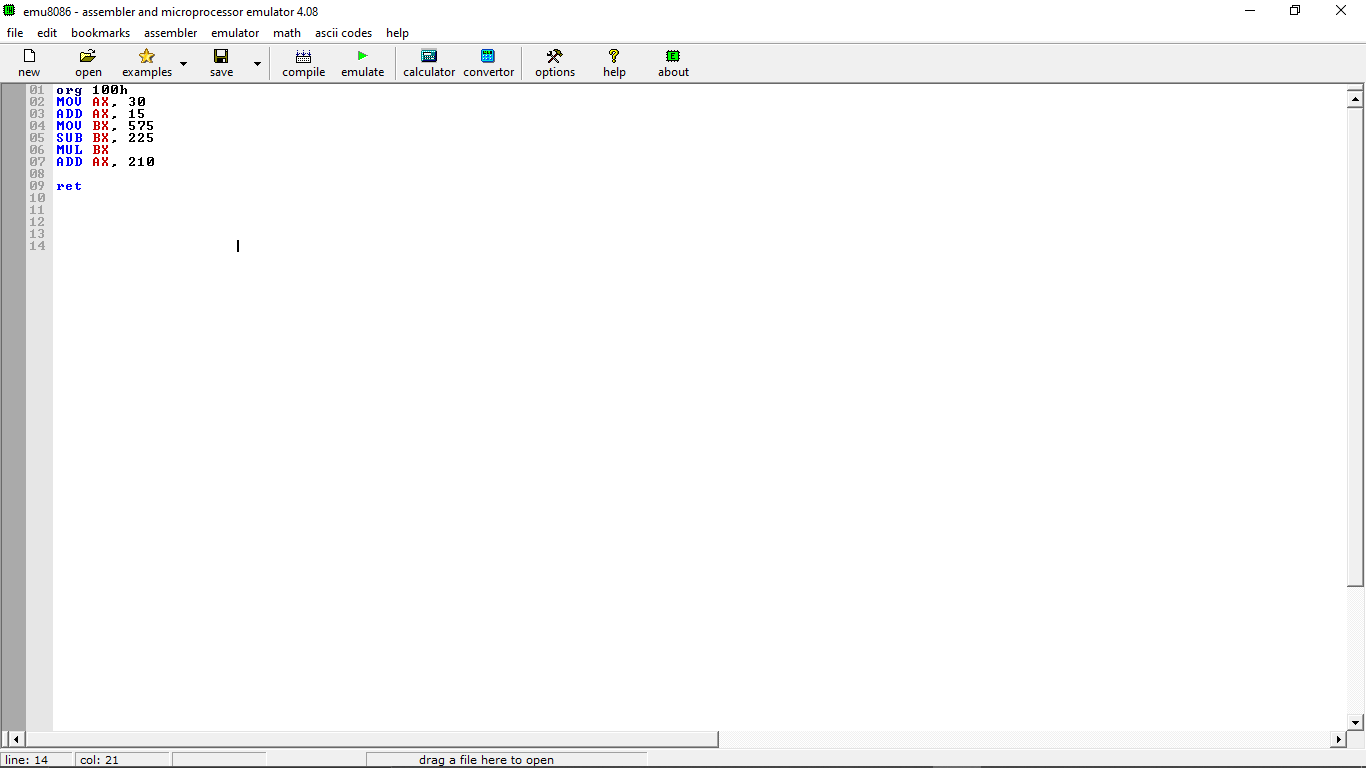
**Exercise – 1.**

**1.Choose “New” and specify “COM template” in emu8086**

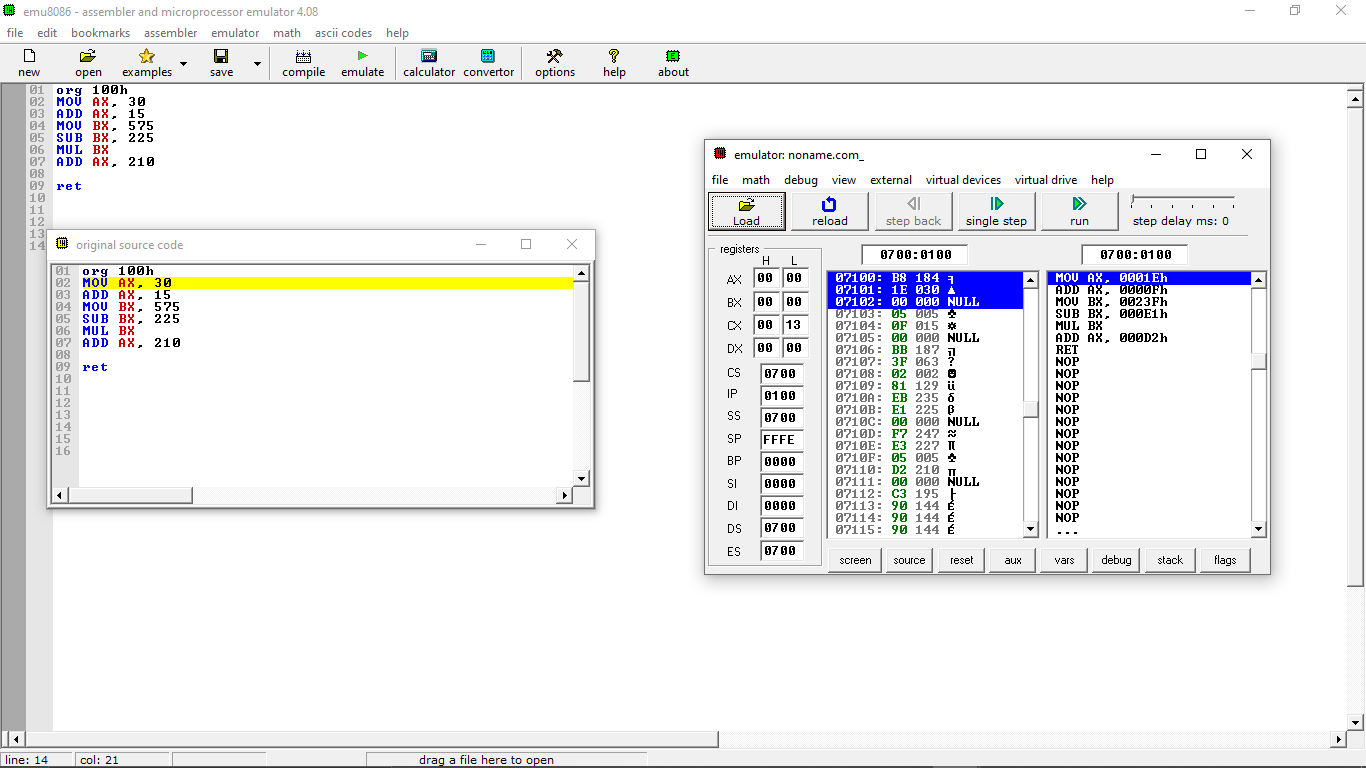




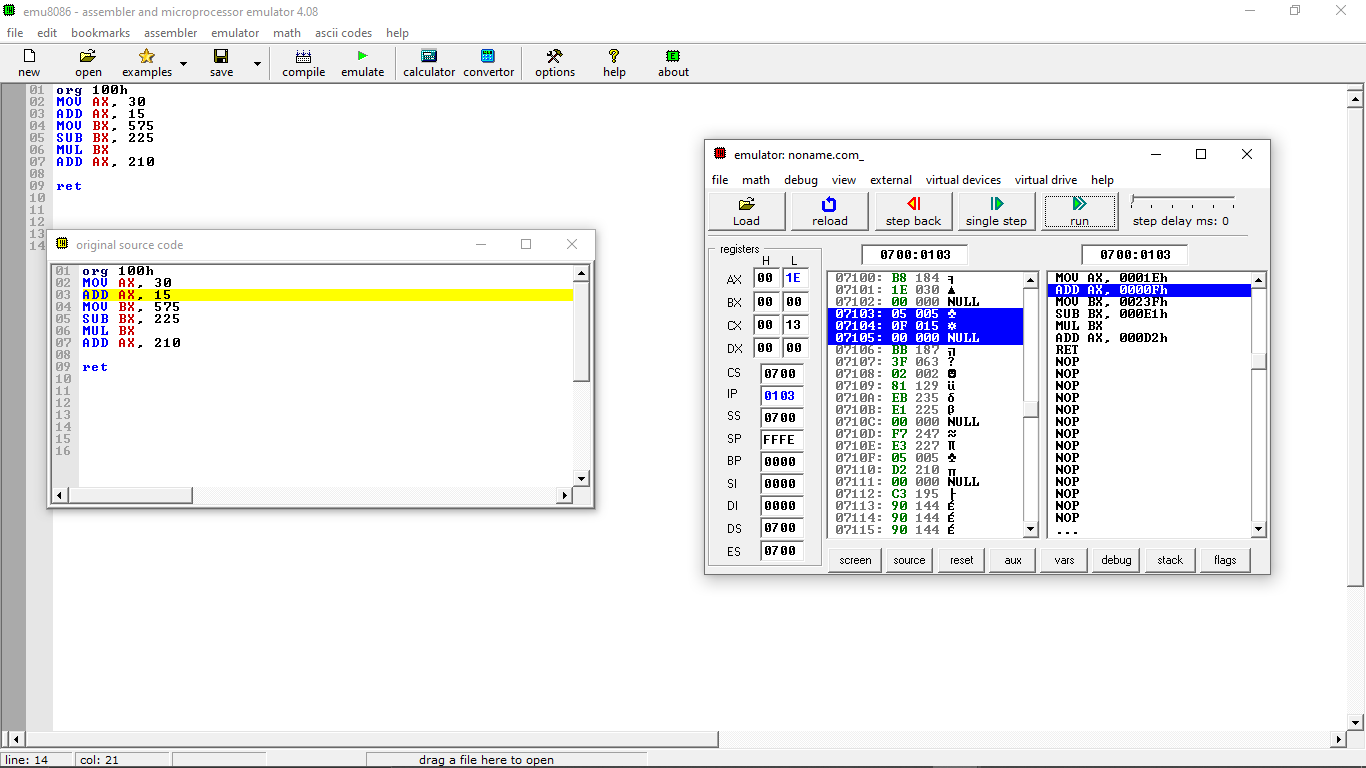
**2.Enter the following code to the editor.**

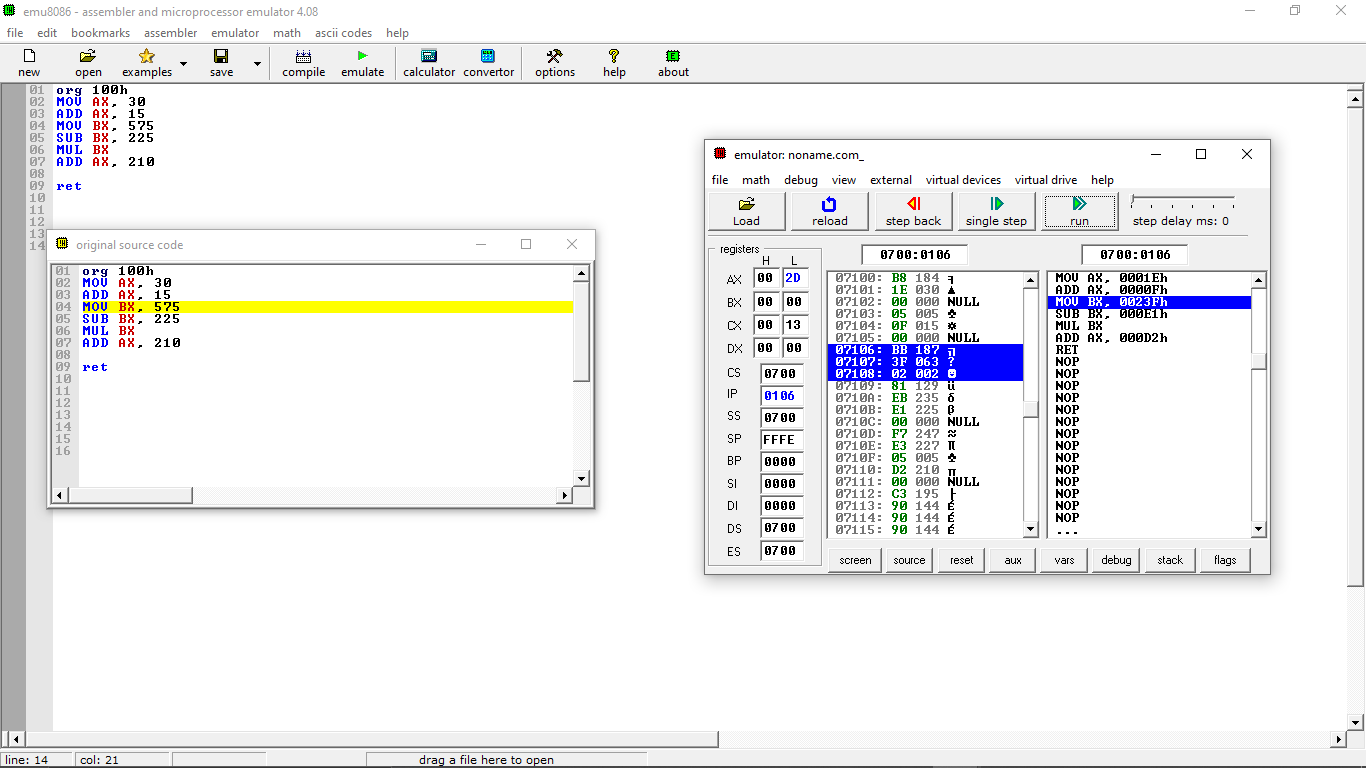


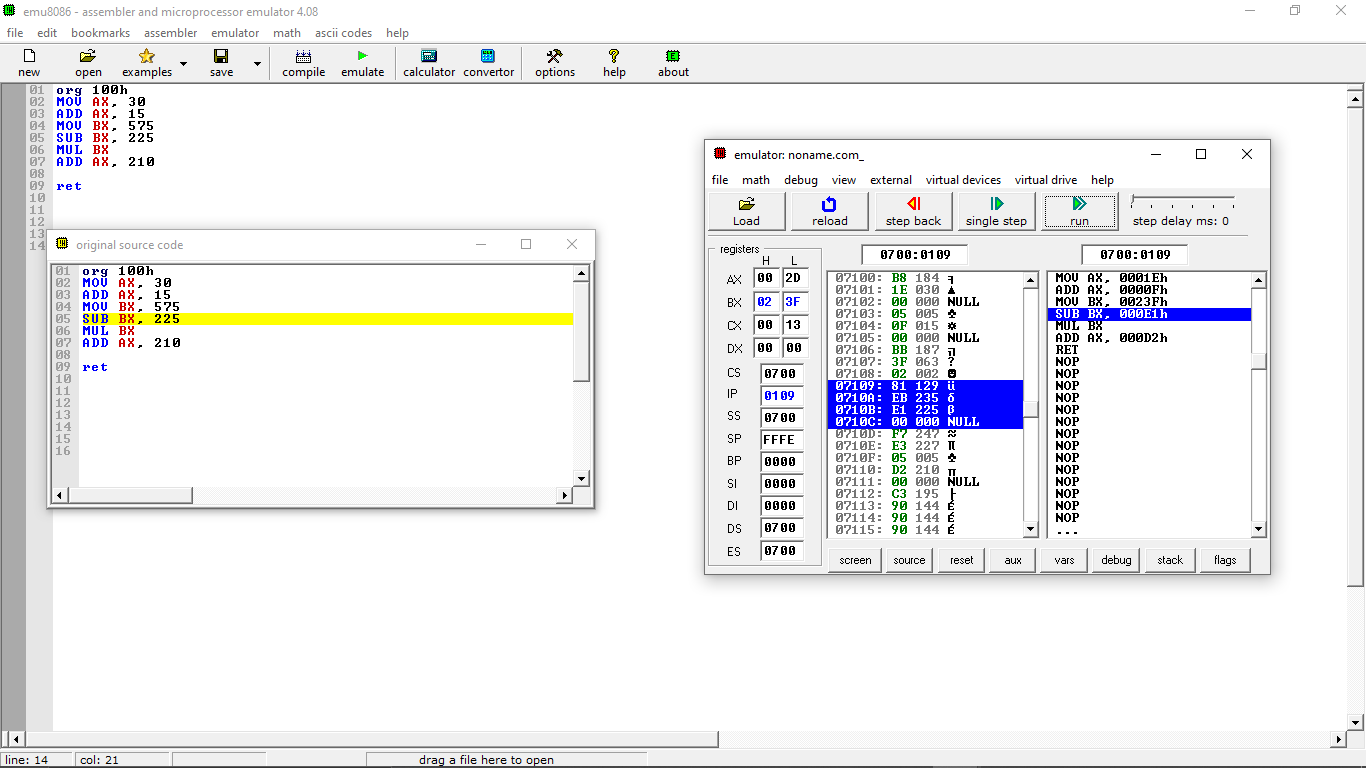
**3.Start emulation by clicking the “emulate” button on the toolbar. A new emulator window will appear.**

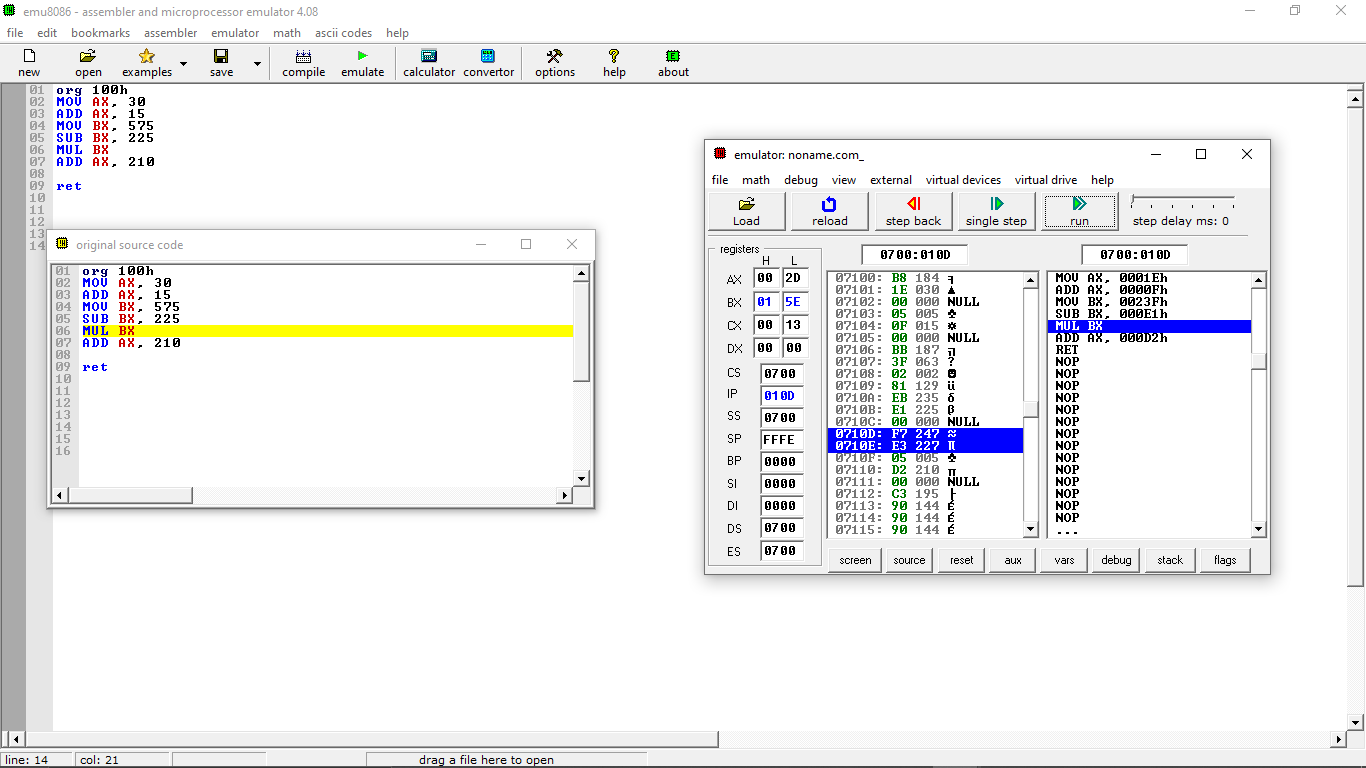


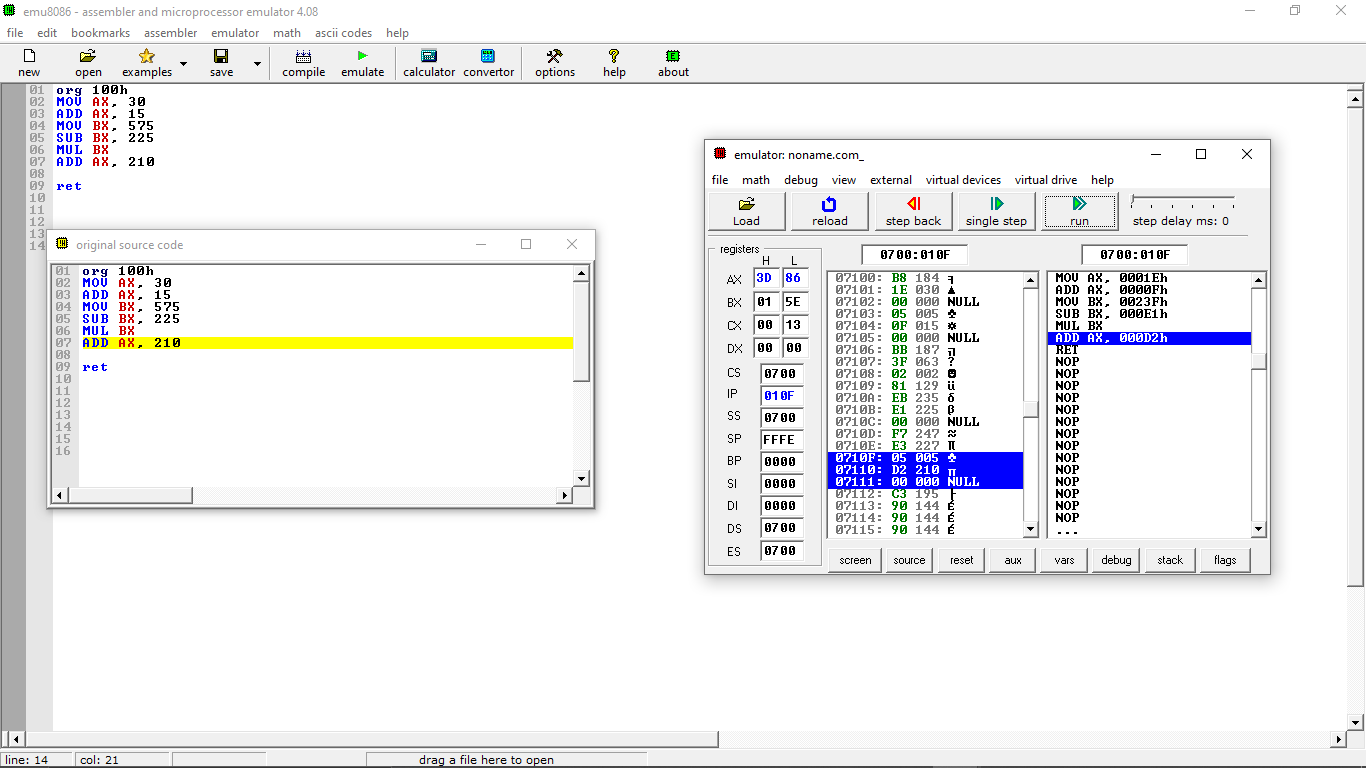
**4.Single-step the program codes by pressing the “single-step” button on the toolbar of the emulator window.**











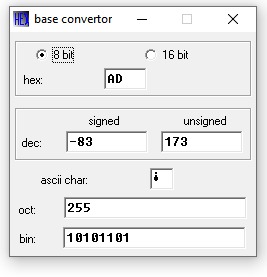
**5.**

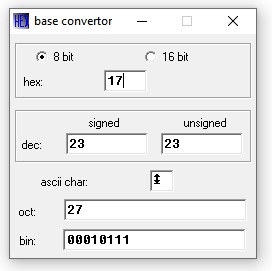
**Each time after pressing the “single-step” button, check and record the contents of AX and BX registers in Table 1.**

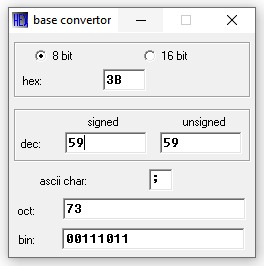
|  |  |  |
| --- | --- | --- |
| **INSTRUCTION** | **AX**  **H L** | **BX**  **H L** |
| MOV AX, 30 | 00 00 | 00 00 |
| ADD AX, 15 | 00 1E | 00 00 |
| MOV BX, 575 | 00 2D | 00 00 |
| SUB BX, 225 | 00 2D | 02 3F |
| MUL BX | 00 2D | 01 5E |
| ADD AX, 210 | 3D 86 | 01 5E |

**Part 5: Exercise Part**

EXERCISE 2







CONCLUSIONS AND RECOMMENDATIONS

THE emulator 8086 is effective in learning assembly programming and we have learnt to appreciate programming with it learning assembler directives and how to code with 8086

REFEREES

Observation of emulator 8086