

Creating & Running a Program

Using Emulator

Programming Steps

Create source program



Editor



.ASM file

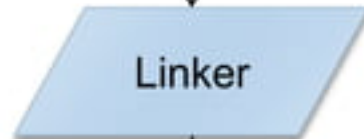


Assembler

Assemble source program



.OBJ file



Linker

Link object program

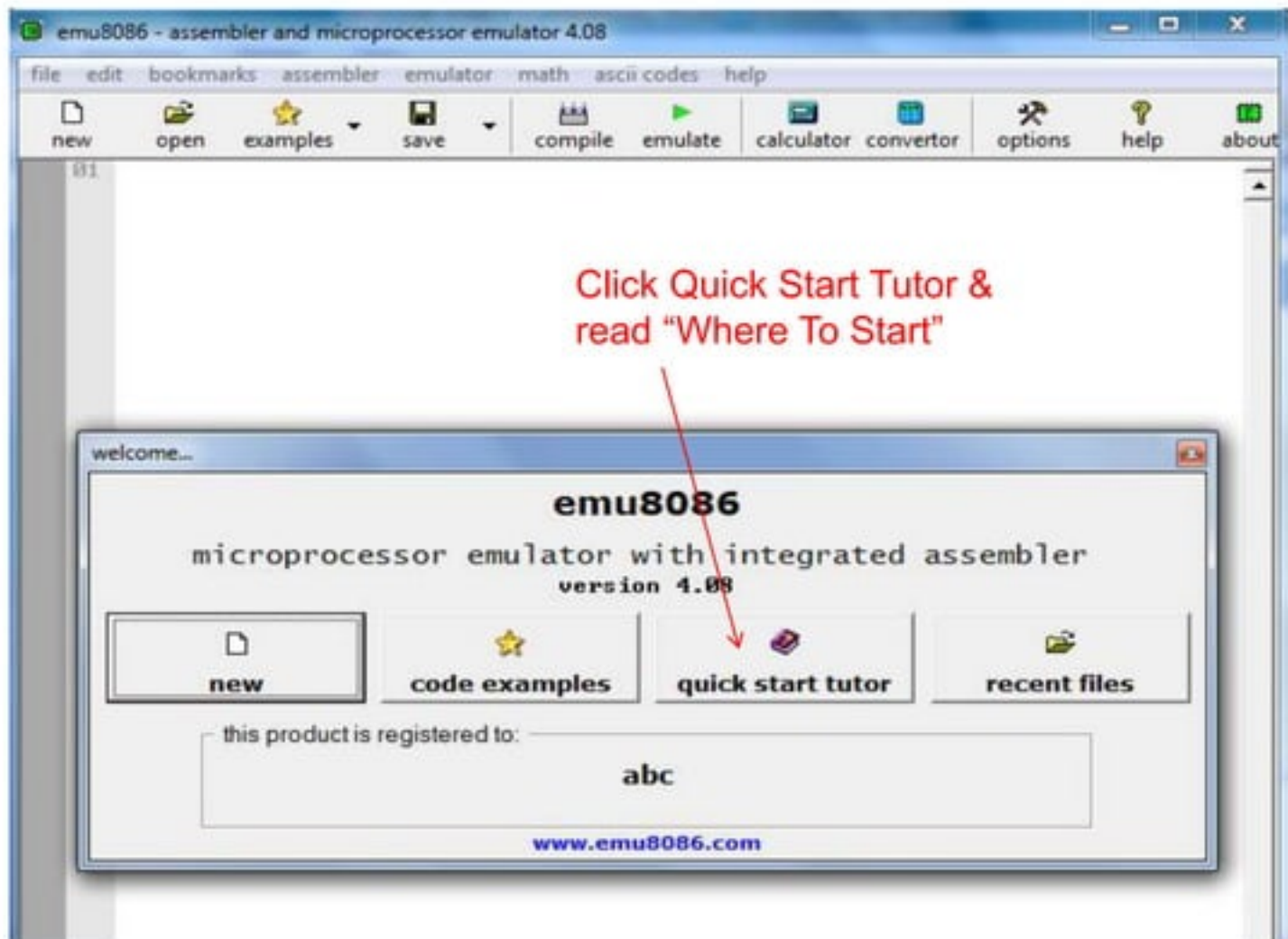


.EXE file

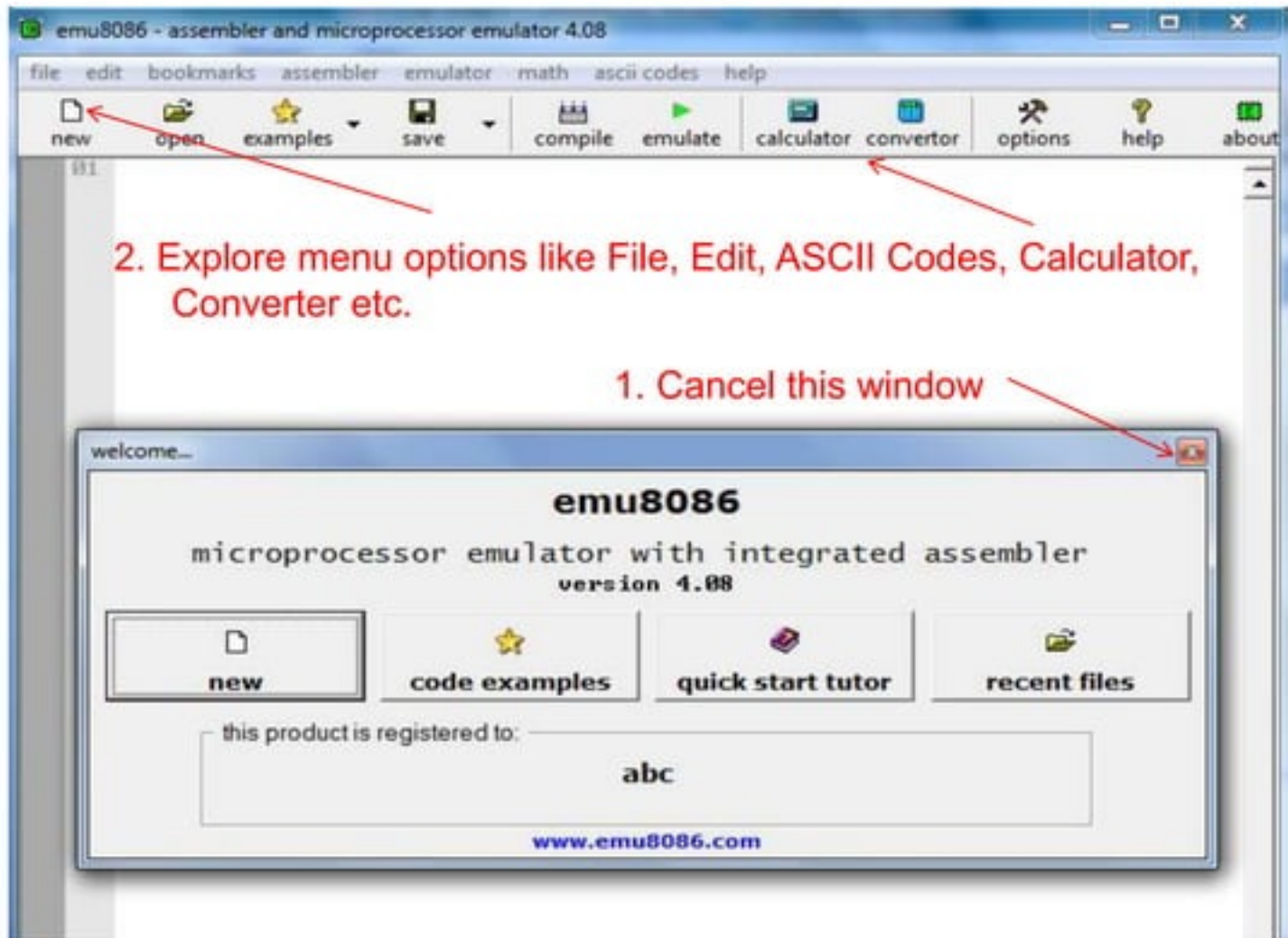
How to Open?

- ▶ Go To Start Menu → Find **emu8086** in Programs
- OR
- ▶ Double click following icon on your desktop:

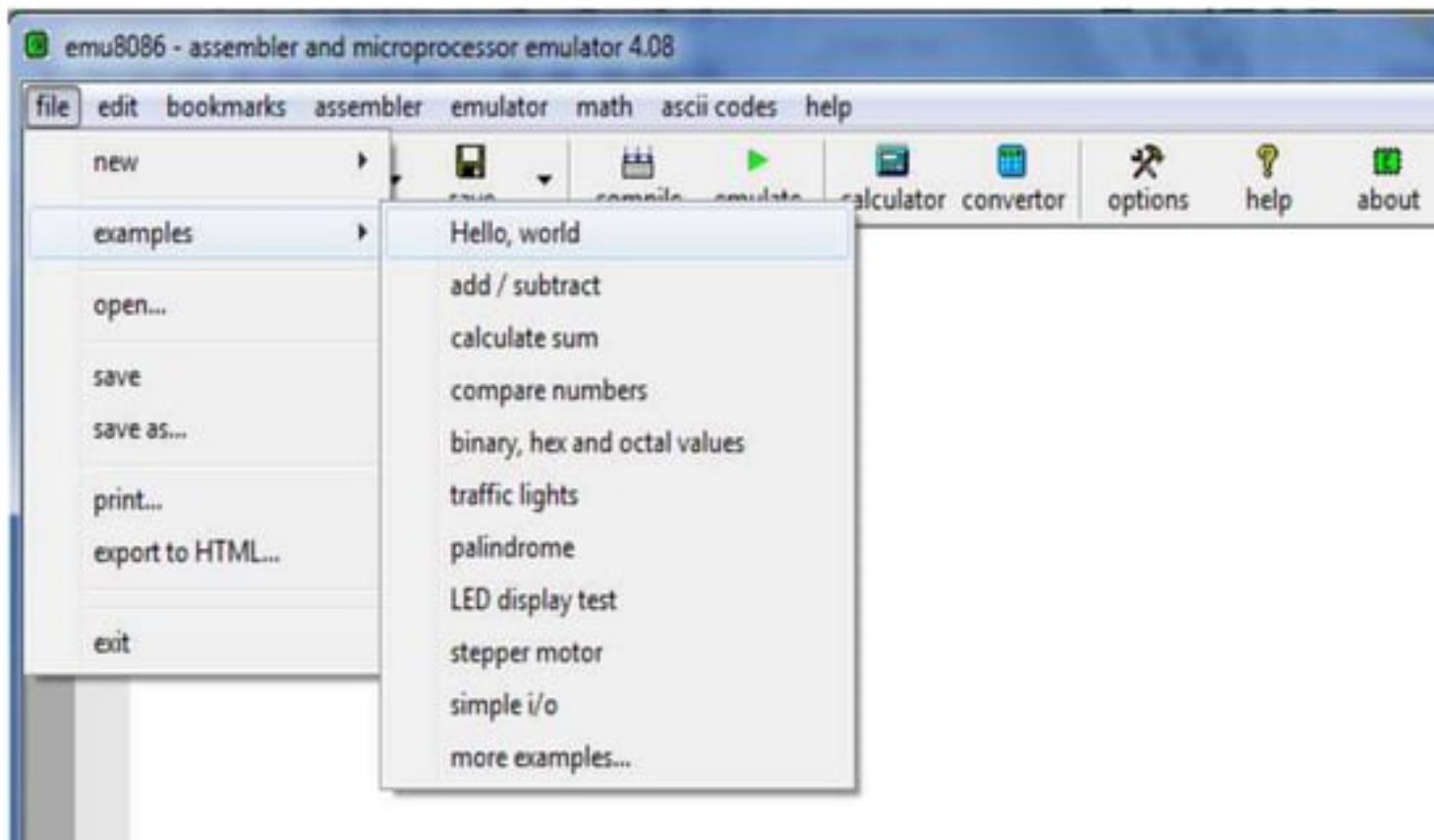




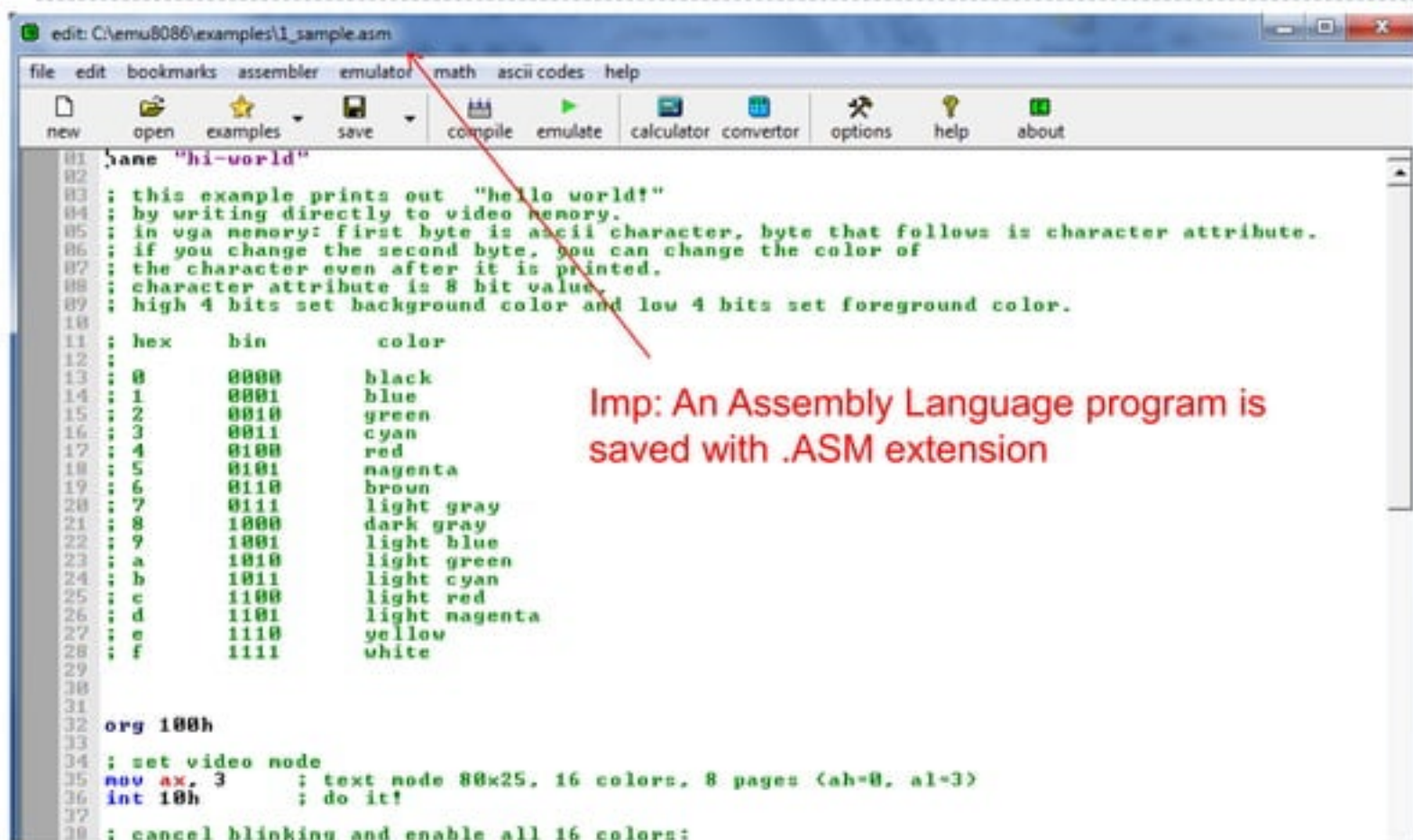
Click Quick Start Tutor &
read "Where To Start"



Open a Program from Examples



Program loaded in Editor



edit: C:\emu8086\examples\1_sample.asm

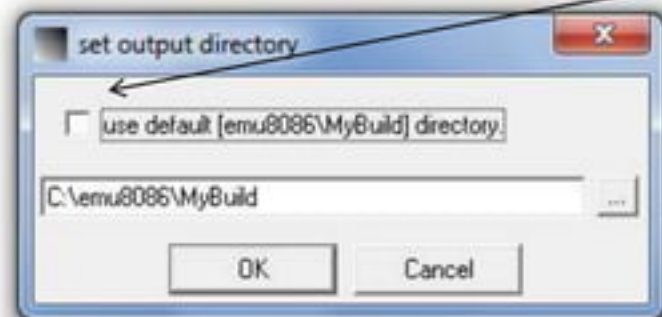
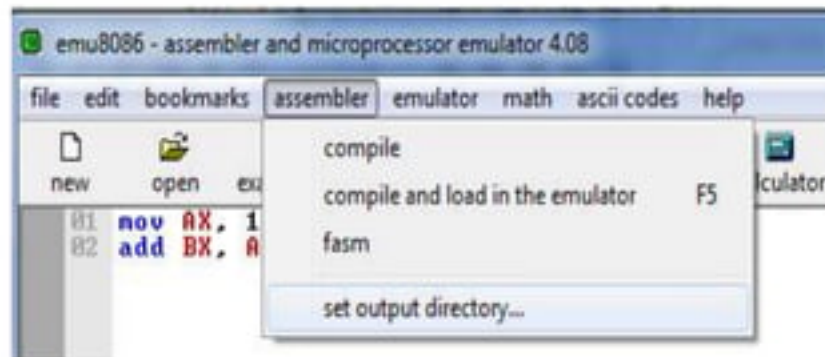
file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
01 name "hi-world"
02
03 ; this example prints out "hello world!"
04 ; by writing directly to video memory.
05 ; in vga memory: first byte is ascii character, byte that follows is character attribute.
06 ; if you change the second byte, you can change the color of
07 ; the character even after it is printed.
08 ; character attribute is 8 bit value.
09 ; high 4 bits set background color and low 4 bits set foreground color.
10
11 ; hex      bin      color
12 ;
13 ; 0        0000     black
14 ; 1        0001     blue
15 ; 2        0010     green
16 ; 3        0011     cyan
17 ; 4        0100     red
18 ; 5        0101     magenta
19 ; 6        0110     brown
20 ; 7        0111     light gray
21 ; 8        1000     dark gray
22 ; 9        1001     light blue
23 ; a        1010     light green
24 ; b        1011     light cyan
25 ; c        1100     light red
26 ; d        1101     light magenta
27 ; e        1110     yellow
28 ; f        1111     white
29
30
31
32 org 100h
33
34 ; set video mode
35 mov ax, 3      ; text mode 80x25, 16 colors, 8 pages (ah=0, al=3)
36 int 10h        ; do it!
37
38 ; cancel blinking and enable all 16 colors:
```

Imp: An Assembly Language program is saved with .ASM extension

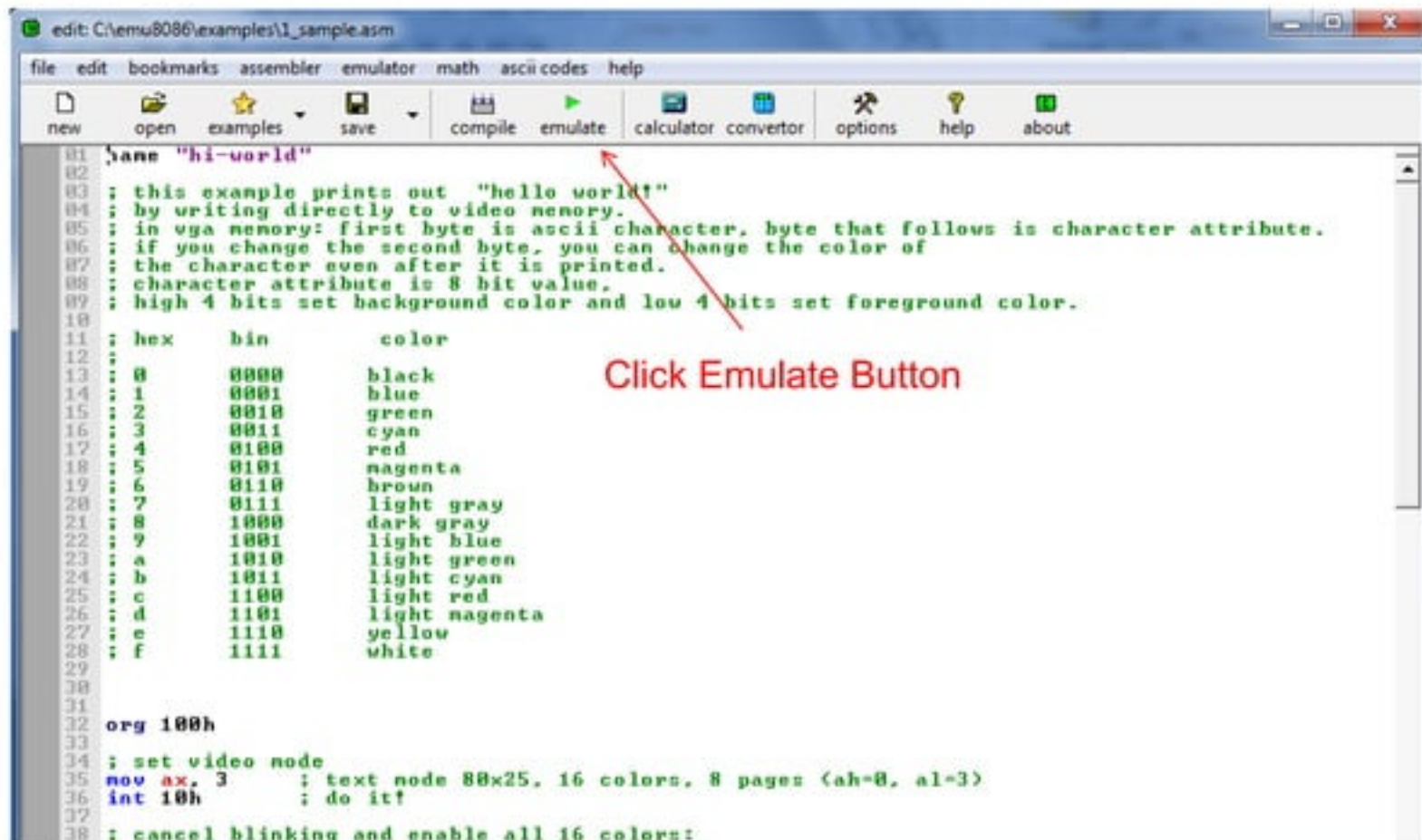
How to change Output Directory



Uncheck this check box

Browse path where executable file will be saved then Click Ok button

How To Run Program



Registers

Logical Address

The screenshot shows the emu8086.com emulator window. At the top is a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, run, and a step delay slider set to 0 ms. Below the toolbar, the 'registers' section on the left lists 16 registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) with their high (H) and low (L) byte values. The 'memory list' section in the center shows a list of memory addresses from 07100 to 07115, with columns for physical address, hex value, decimal value, and ASCII value. The 'disassembled machine code' section on the right shows the corresponding assembly instructions for the selected memory range. At the bottom, there are buttons for screen, source, reset, aux, vars, debug, stack, and flags.

Register	H	L
AX	00	00
BX	00	00
CX	00	40
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

Physical Address	Hex	Decimal	ASCII
07100	B8	184	
07101	03	003	
07102	00	000	NULL
07103	CD	205	=
07104	10	016	
07105	B8	184	
07106	03	003	
07107	10	016	
07108	BB	187	
07109	00	000	NULL
0710A	00	000	NULL
0710B	CD	205	=
0710C	10	016	
0710D	B2	178	
0710E	00	000	NULL
0710F	B6	182	
07110	00	000	NULL
07111	B3	179	
07112	00	000	NULL
07113	EB	235	
07114	09	009	TAB
07115	FE	254	

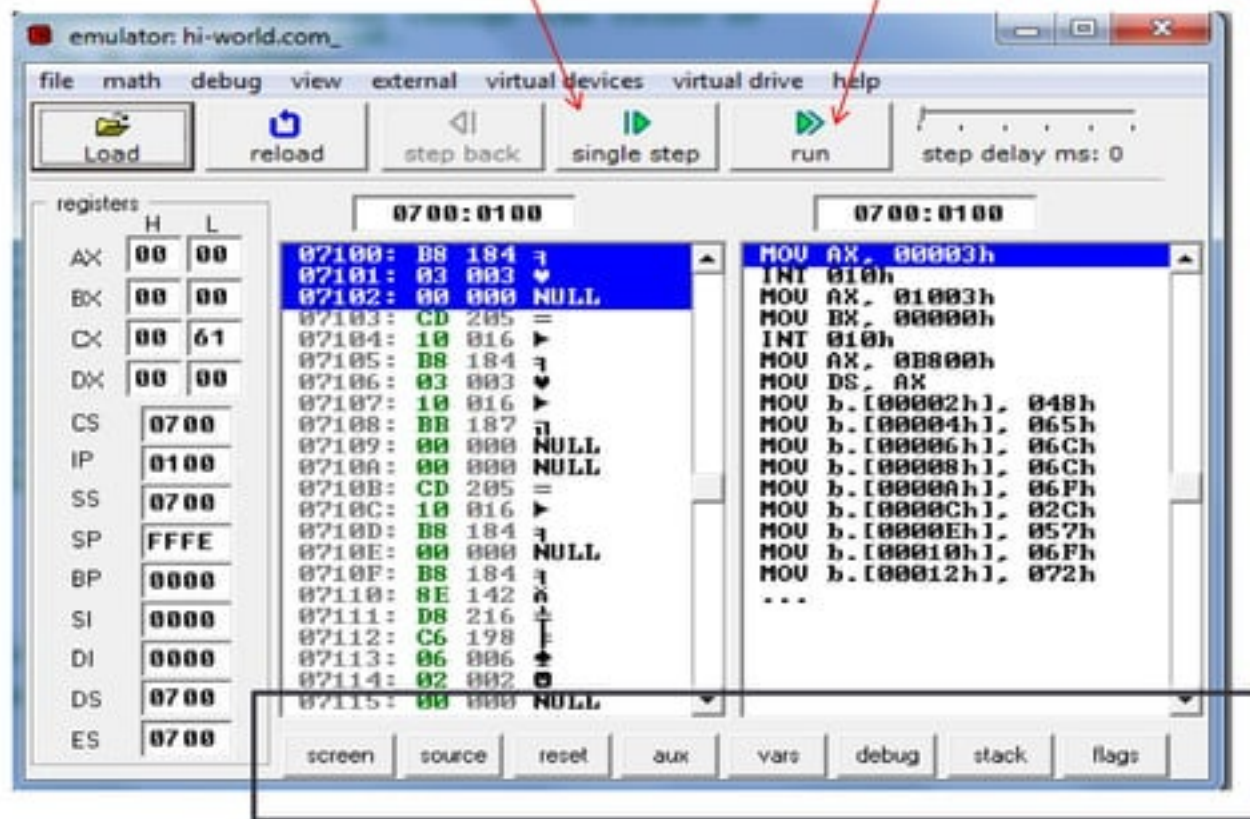
Disassembled Machine Code
MOV AX, 00003h
INT 010h
MOV AX, 01003h
MOV BX, 00000h
INT 010h
MOV DL, 00h
MOV DH, 00h
MOV BL, 00h
JMP 011Eh
INC DH
CMP DH, 010h
JZ 0138h
MOV DL, 00h
MOV AH, 02h
INT 010h
MOV AL, 061h
MOV BH, 00h
MOV CX, 00001h
MOV AH, 09h
INT 010h
INC BL
...

Physical Address: **HEX** **DECIMAL** **ASCII**

The Memory List

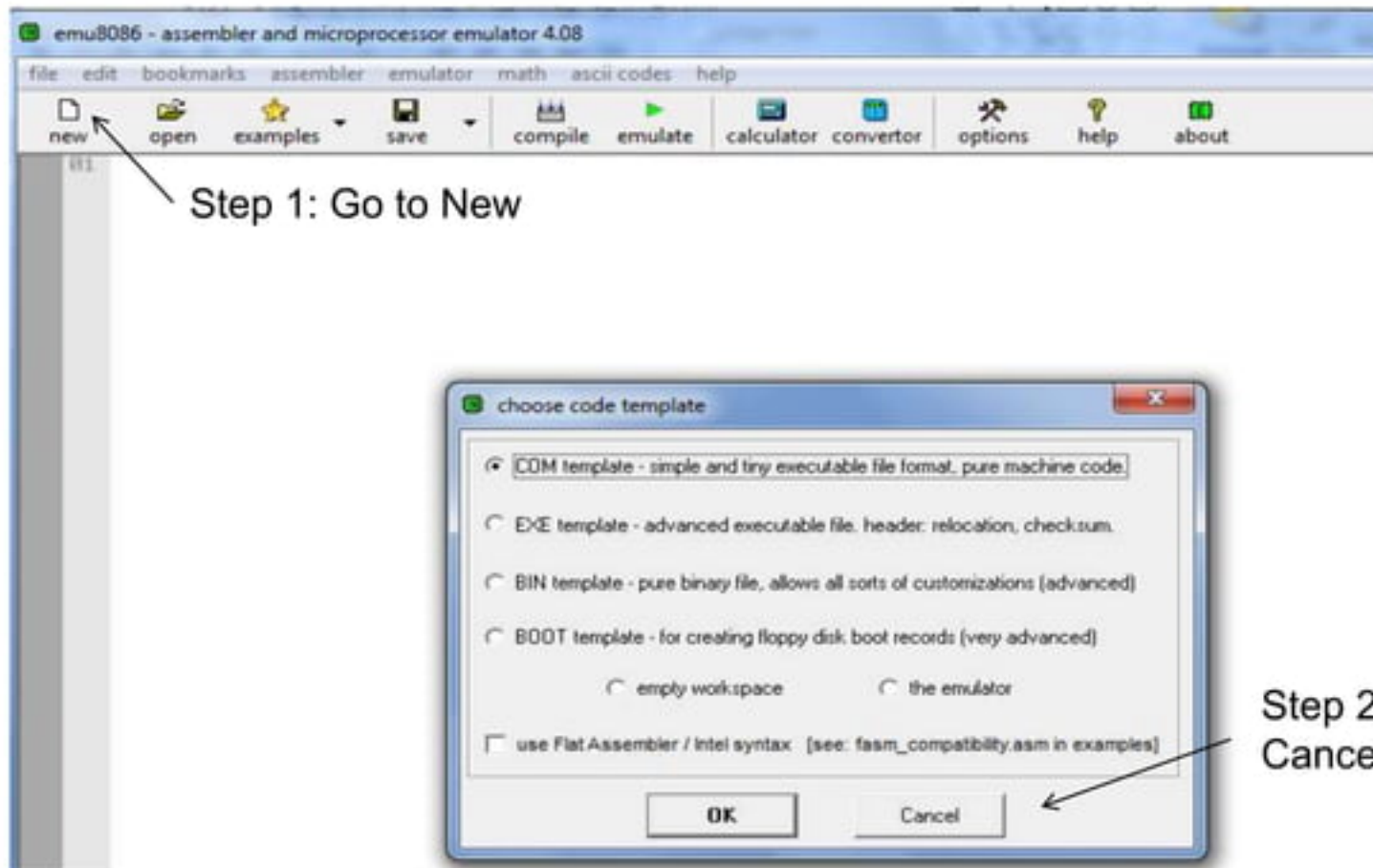
Single Step: To execute instruction one by one i.e. stop after each instruction

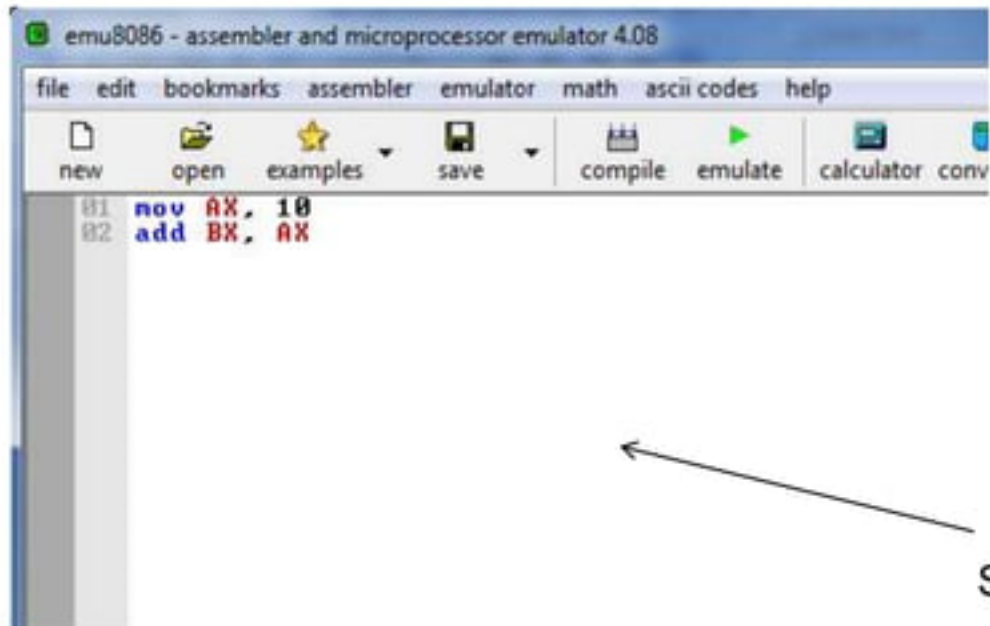
Run: To run complete program



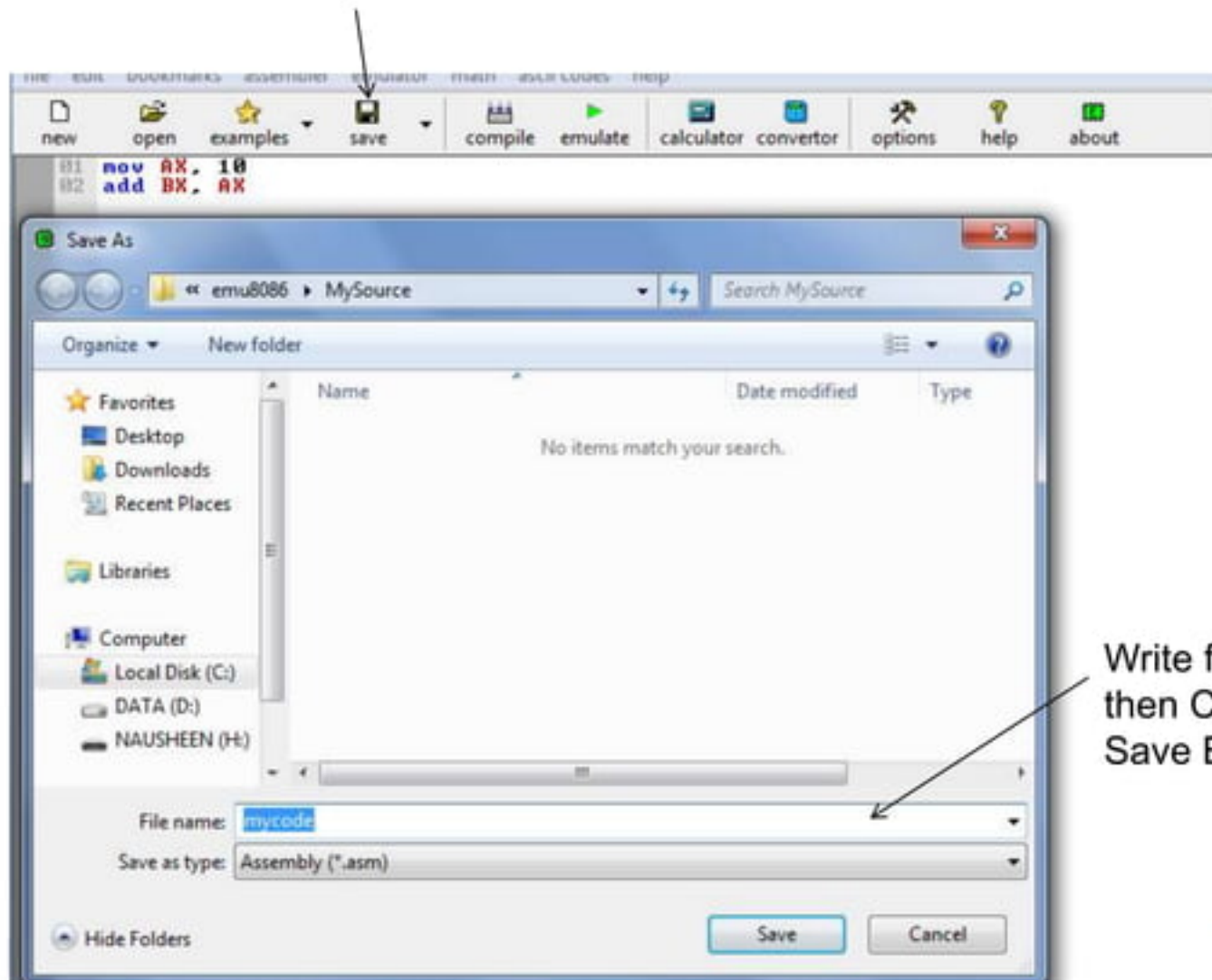
Explore what these buttons display!

How To Create a New File

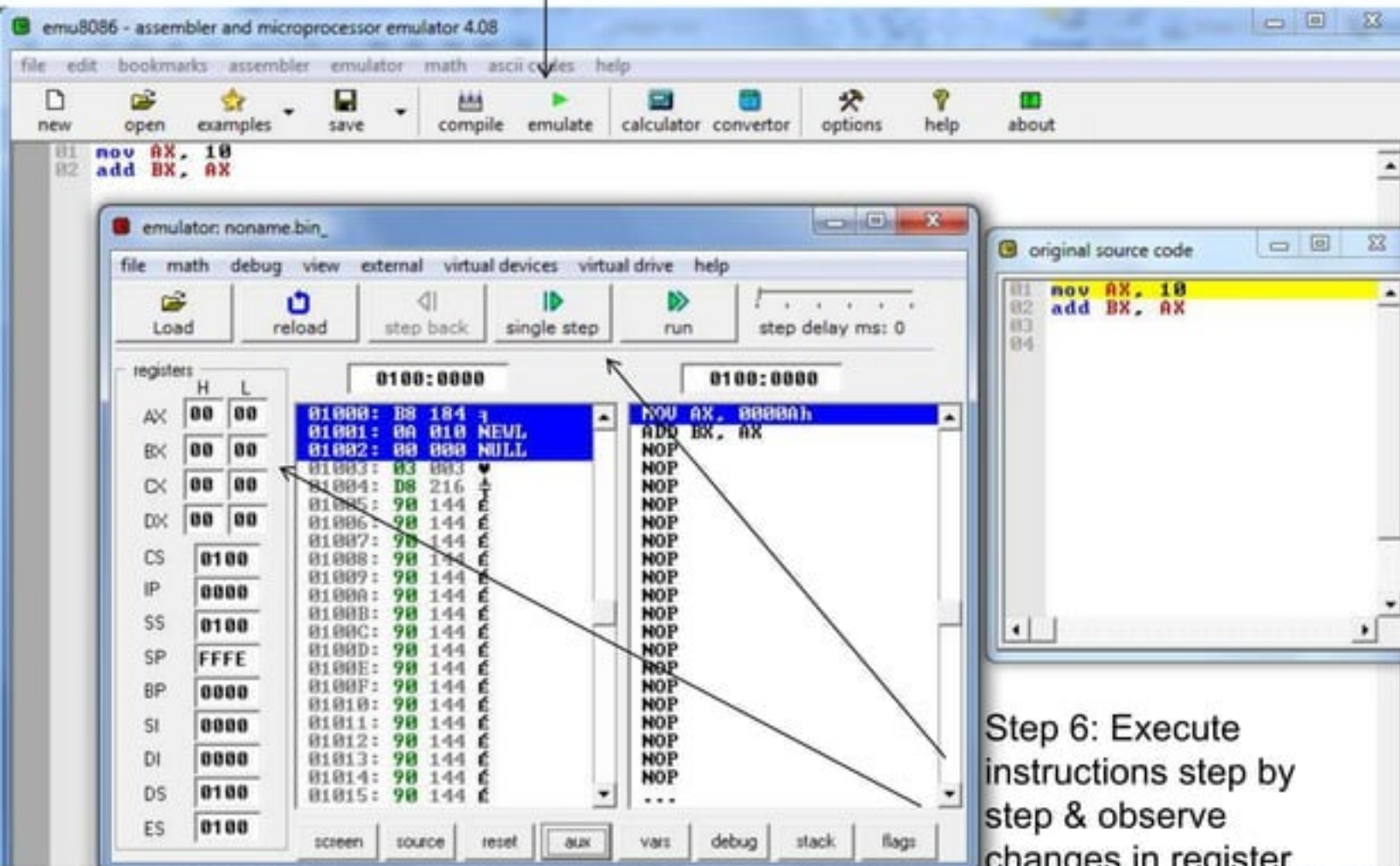




Step 4: Save your code



Step 5: Emulate Program



```

01  ORG 100h
02  .DATA
03  MSG DB "HELLO", 0Ah, 0Dh, "$"
04  MSG1 DW "HELLO", 0Ah, 0Dh, "$"
05  .CODE

```

Random Access Memory

0700:0102

update

☐ table

☒ list

0700:0102:	48	072	H
0700:0103:	45	069	E
0700:0104:	4C	076	L
0700:0105:	4C	076	L
0700:0106:	4F	079	O
0700:0107:	0A	010	NEWL
0700:0108:	0D	013	CRET
0700:0109:	24	036	\$
0700:010A:	48	072	H
0700:010B:	45	069	E
0700:010C:	4C	076	L
0700:010D:	4C	076	L
0700:010E:	4F	079	O
0700:010F:	00	000	NULL
0700:0110:	0A	010	NEWL
0700:0111:	00	000	NULL
0700:0112:	0D	013	CRET
0700:0113:	00	000	NULL
0700:0114:	24	036	\$