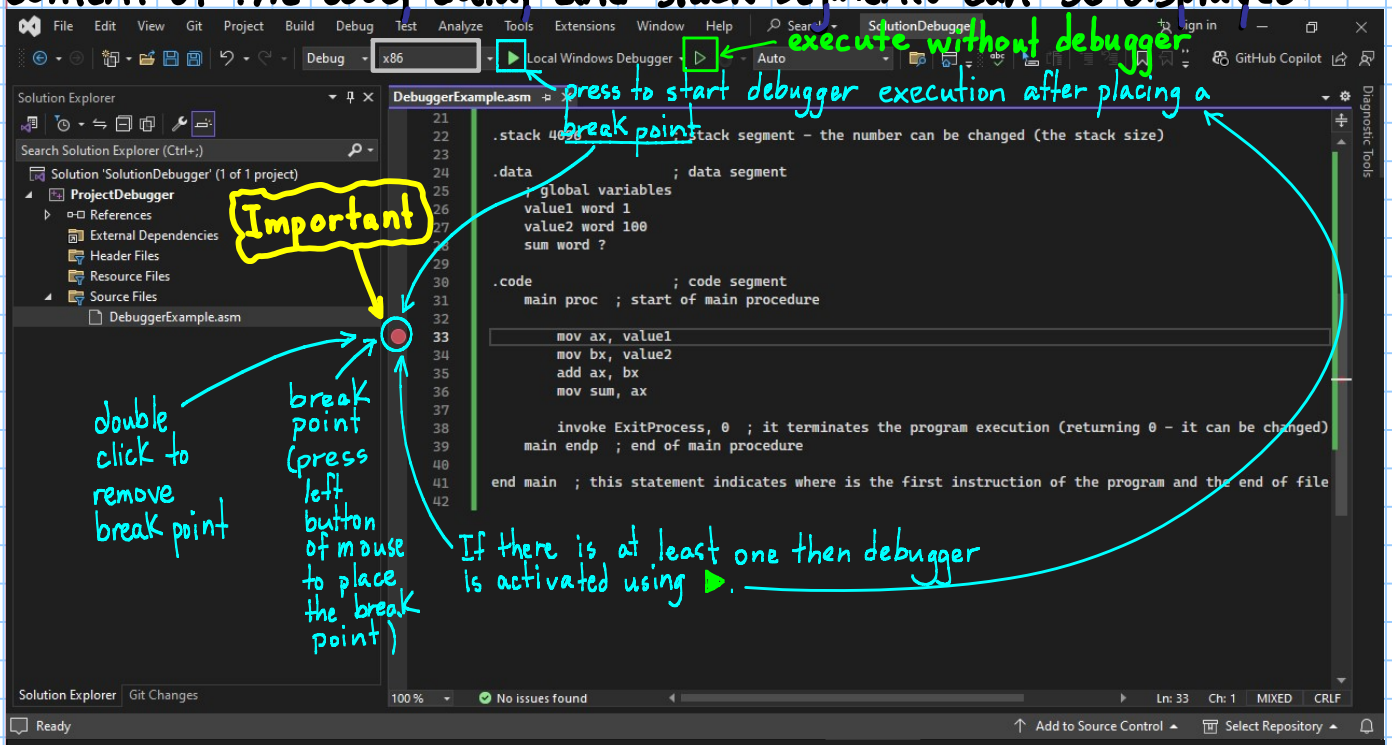
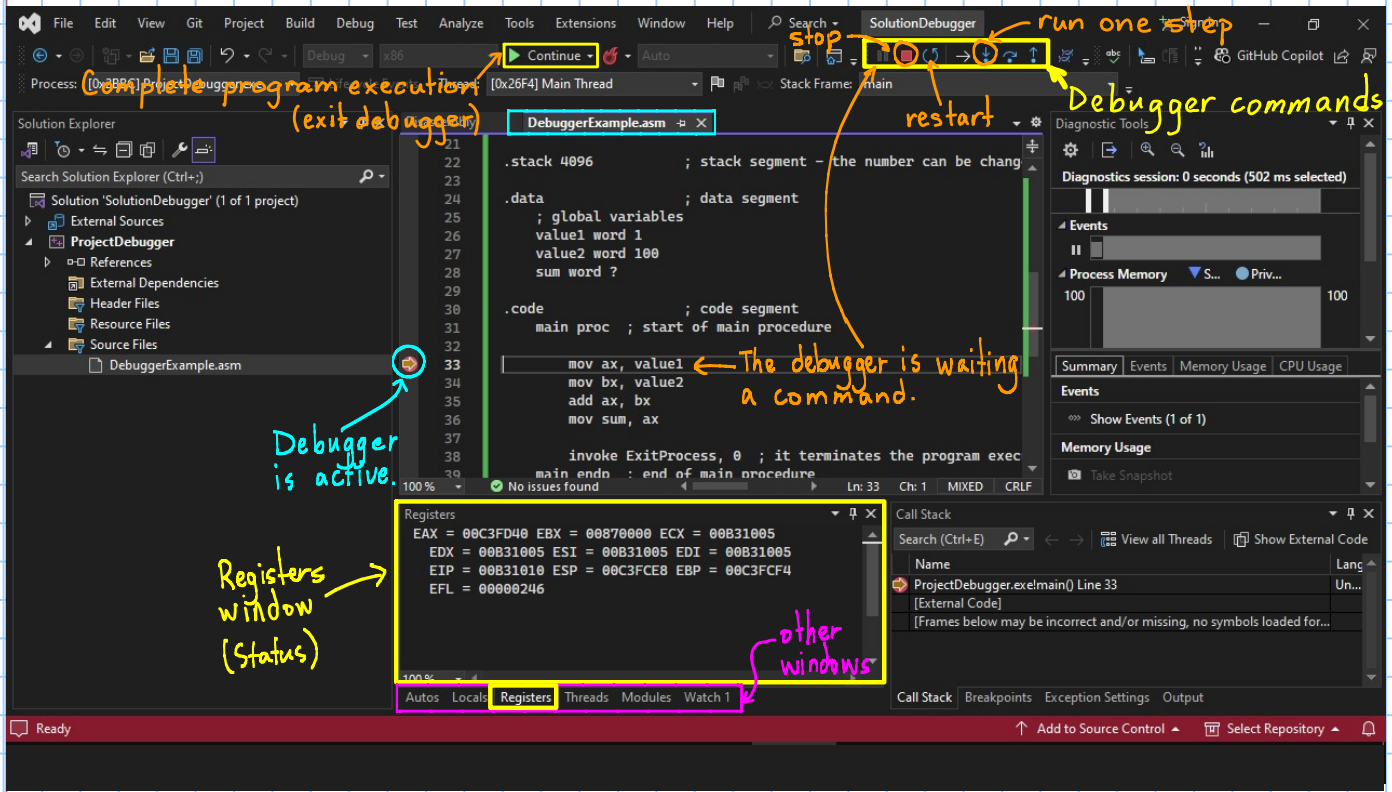
How to use the debugger to run a program

Note Title

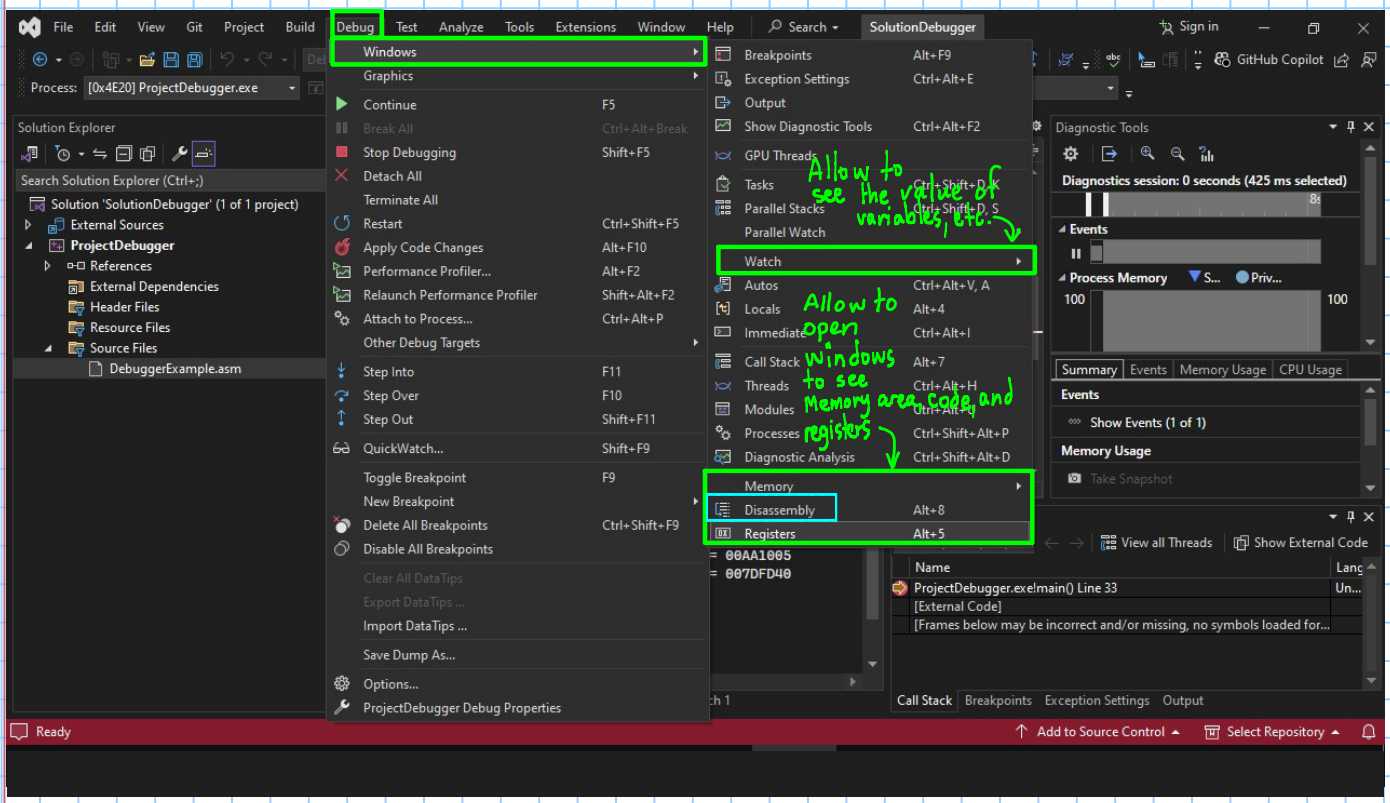
Start debugger execution (the images are for X86; similar for X64): Prof. L.Ortiz

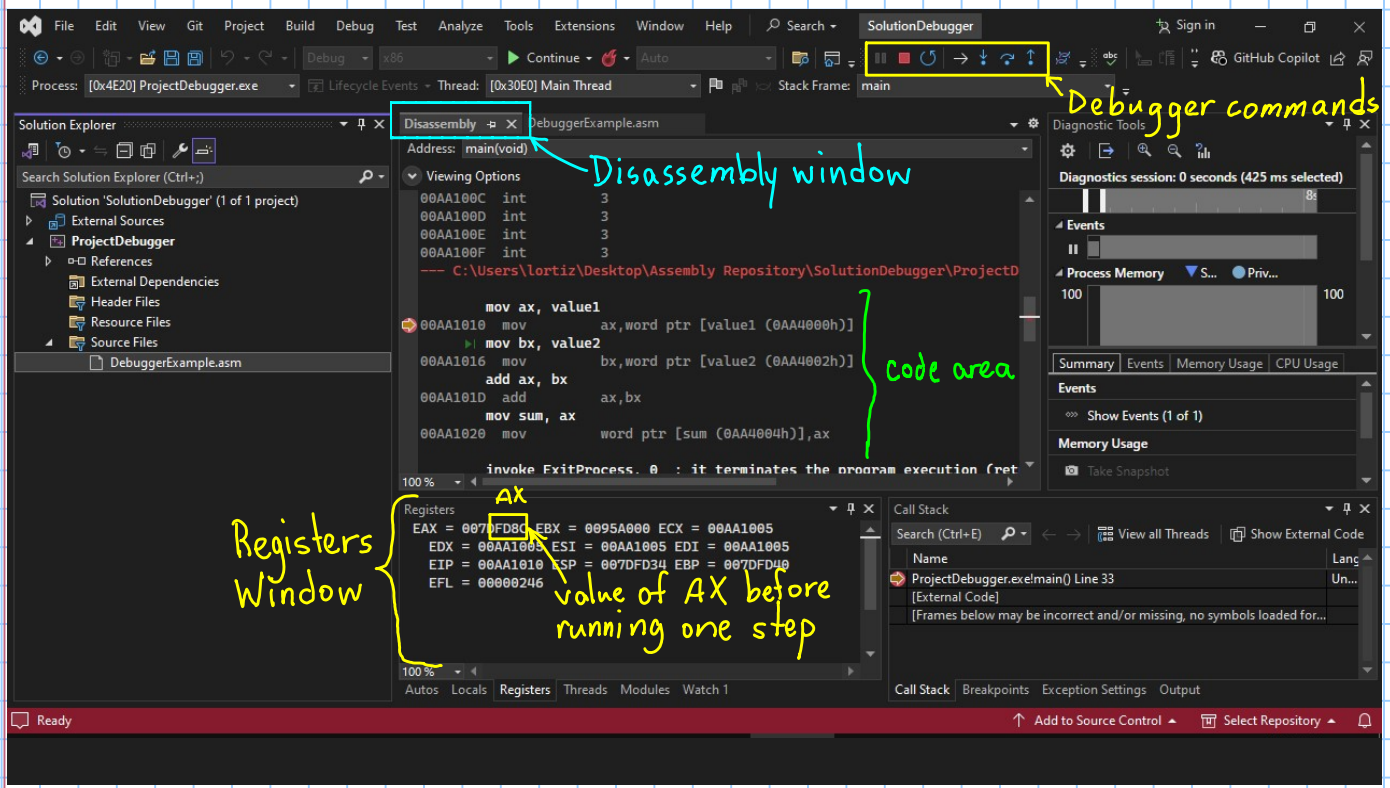
It is used to show the execution of each instruction of a program. In addition, it can be used to show the result of an instruction or the program by displaying the value of the registers and variables. The content of the code, data, and stack segments can be displayed.



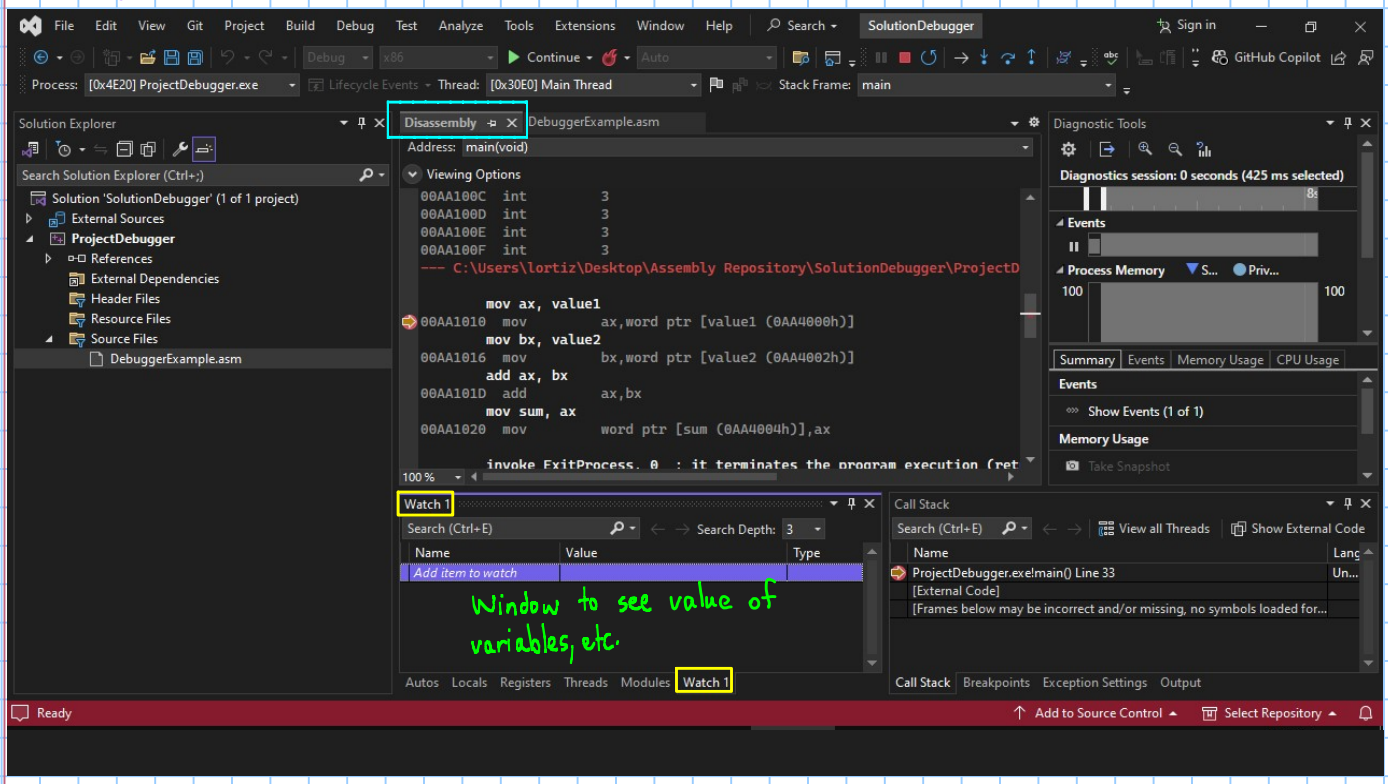


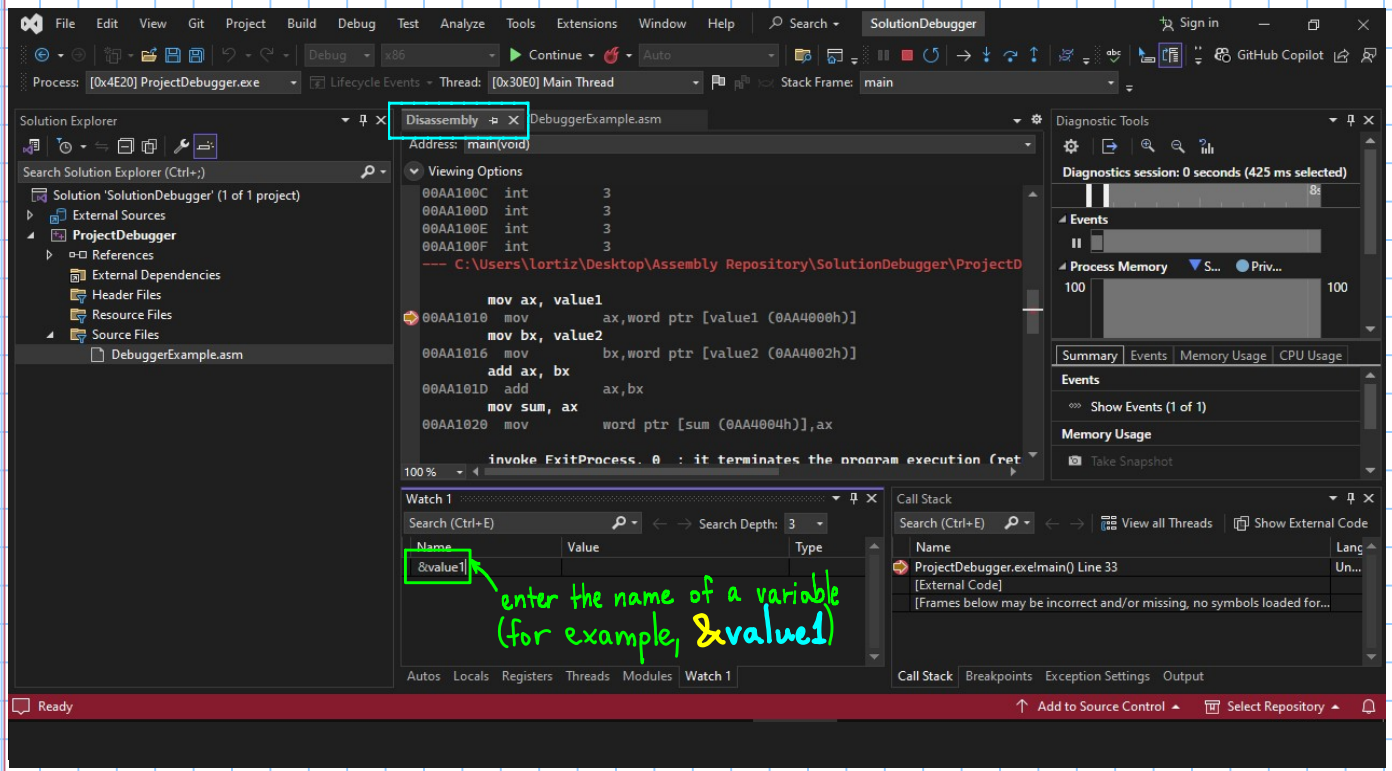
Windows for showing memory area, registers, and machine code

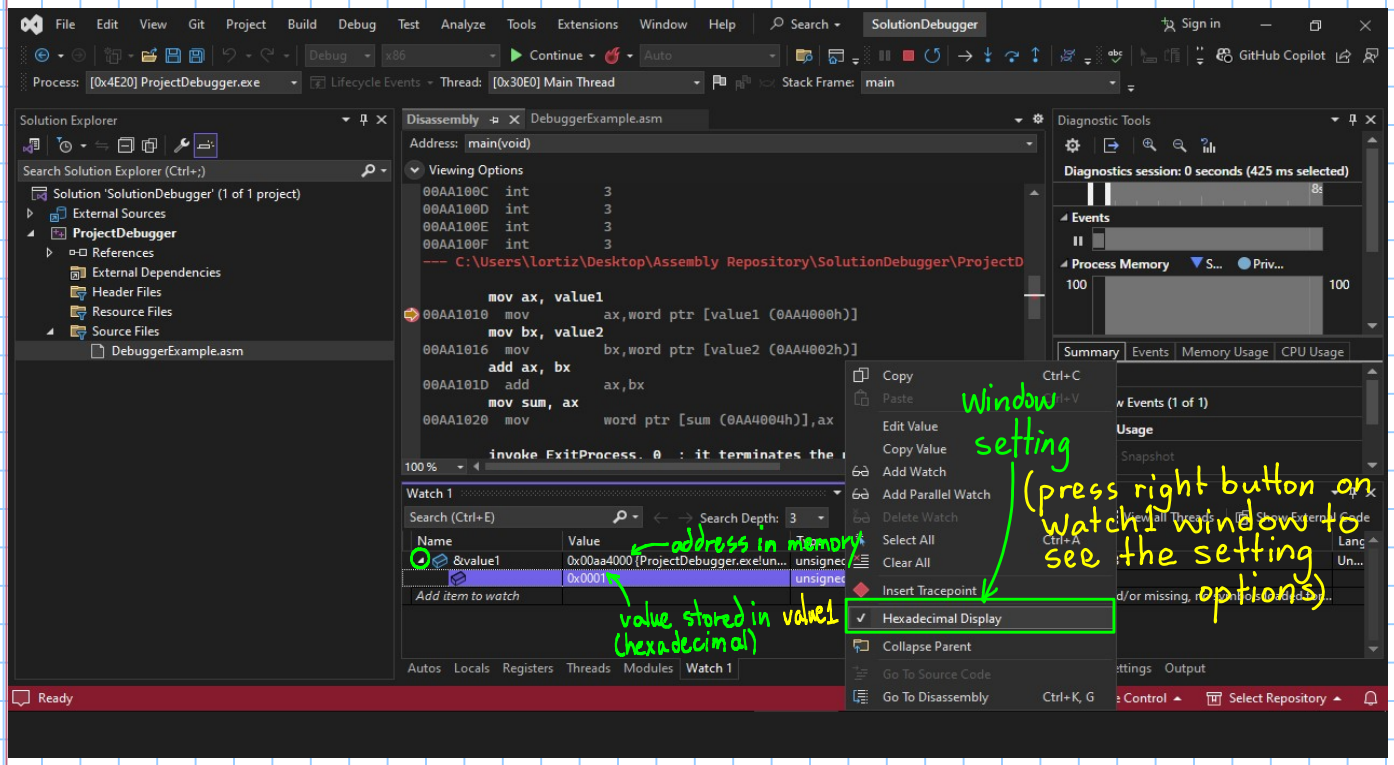




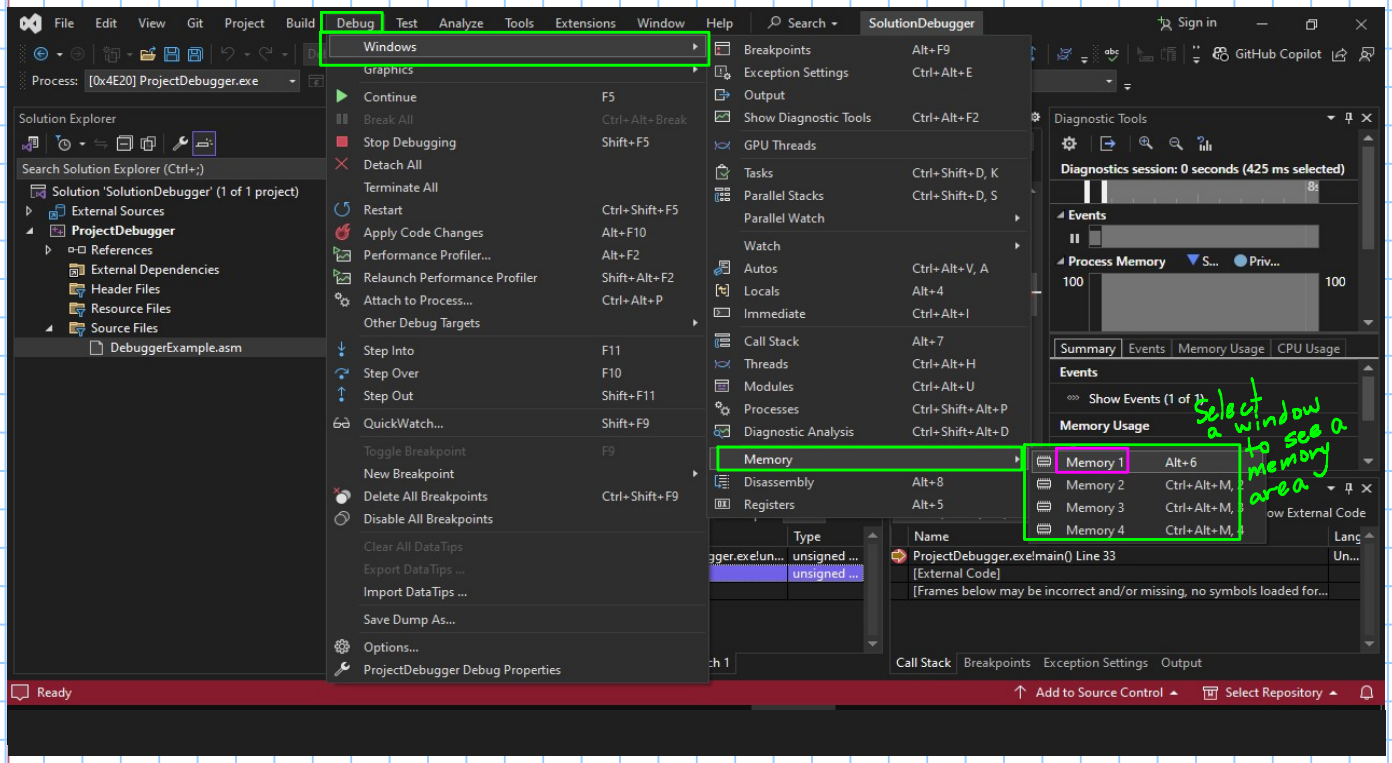
Show global variables in data segment

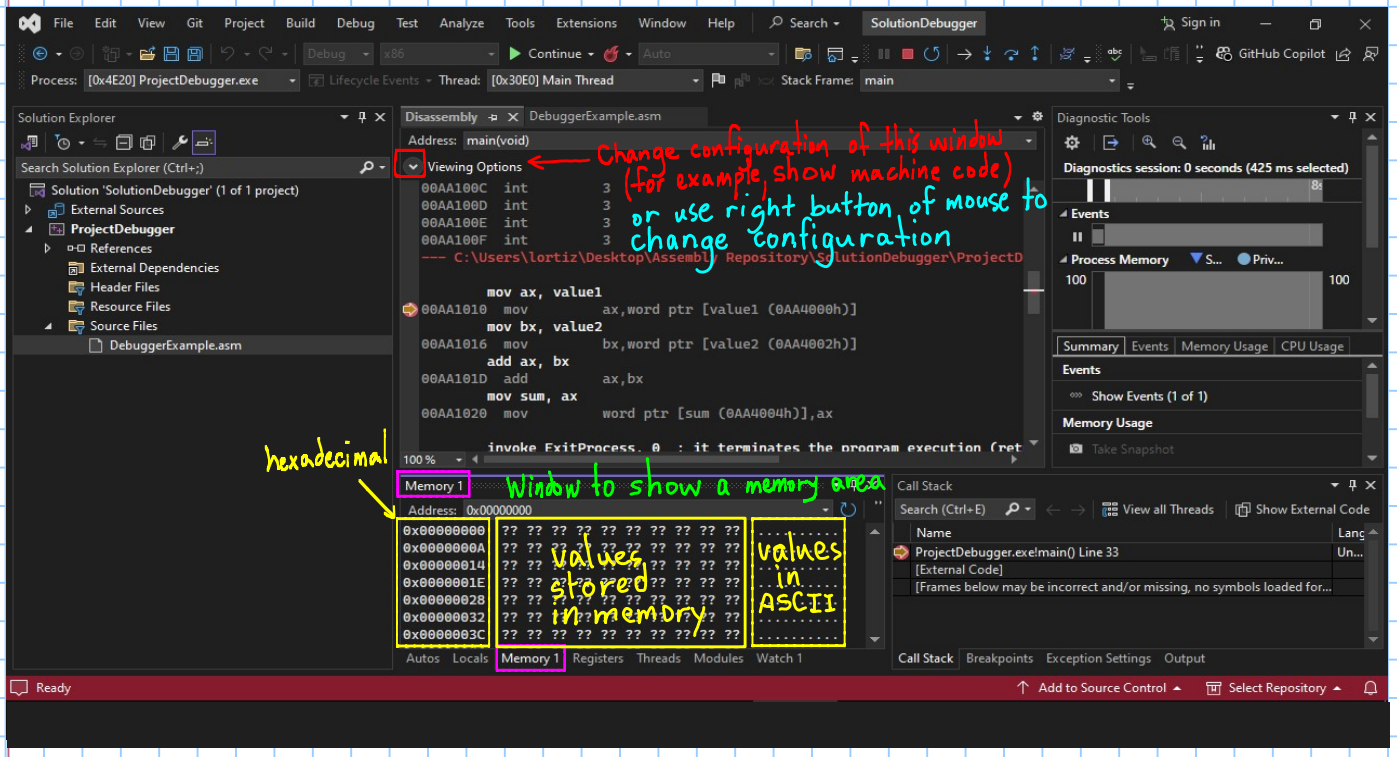




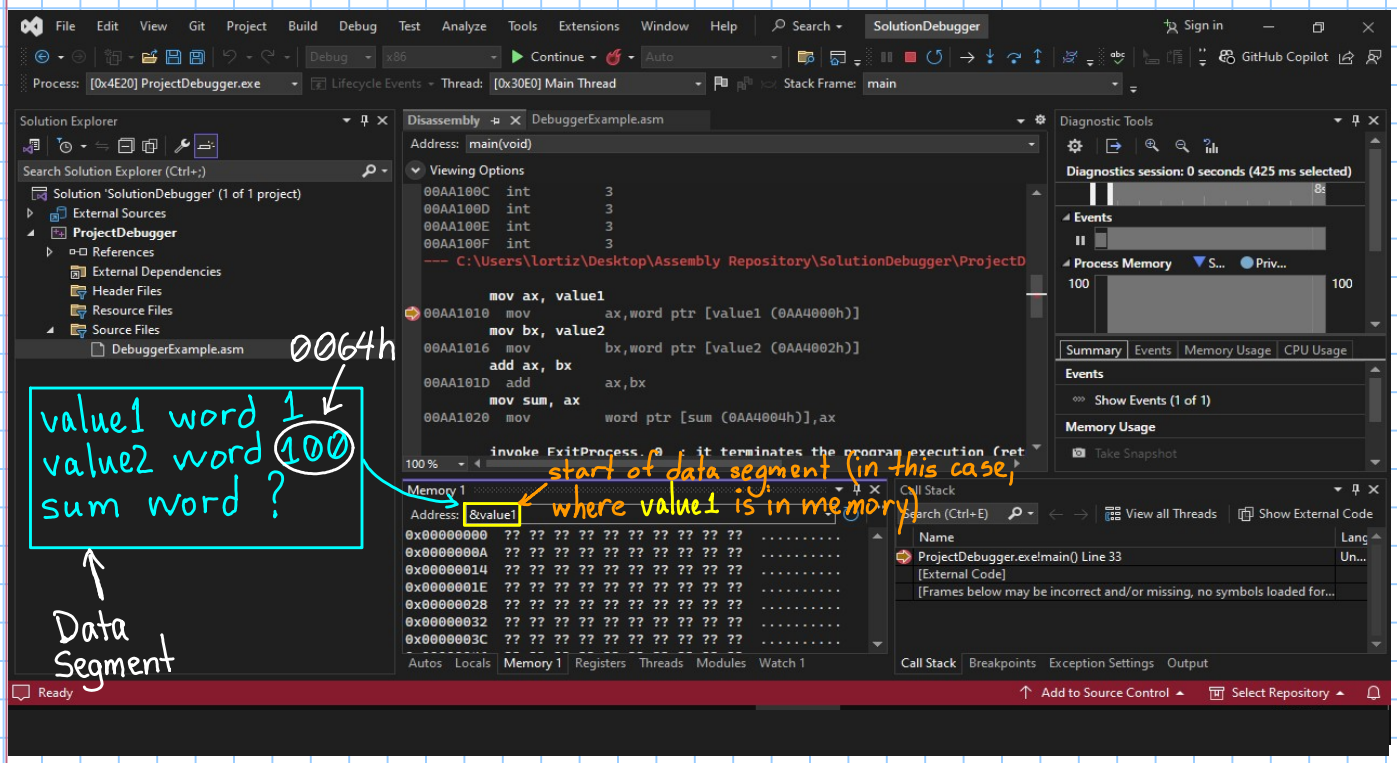

Window setting
(press right button on watch1 window to see the setting options)
address in memory
value stored in value1 (hexadecimal)
Hexadecimal Display

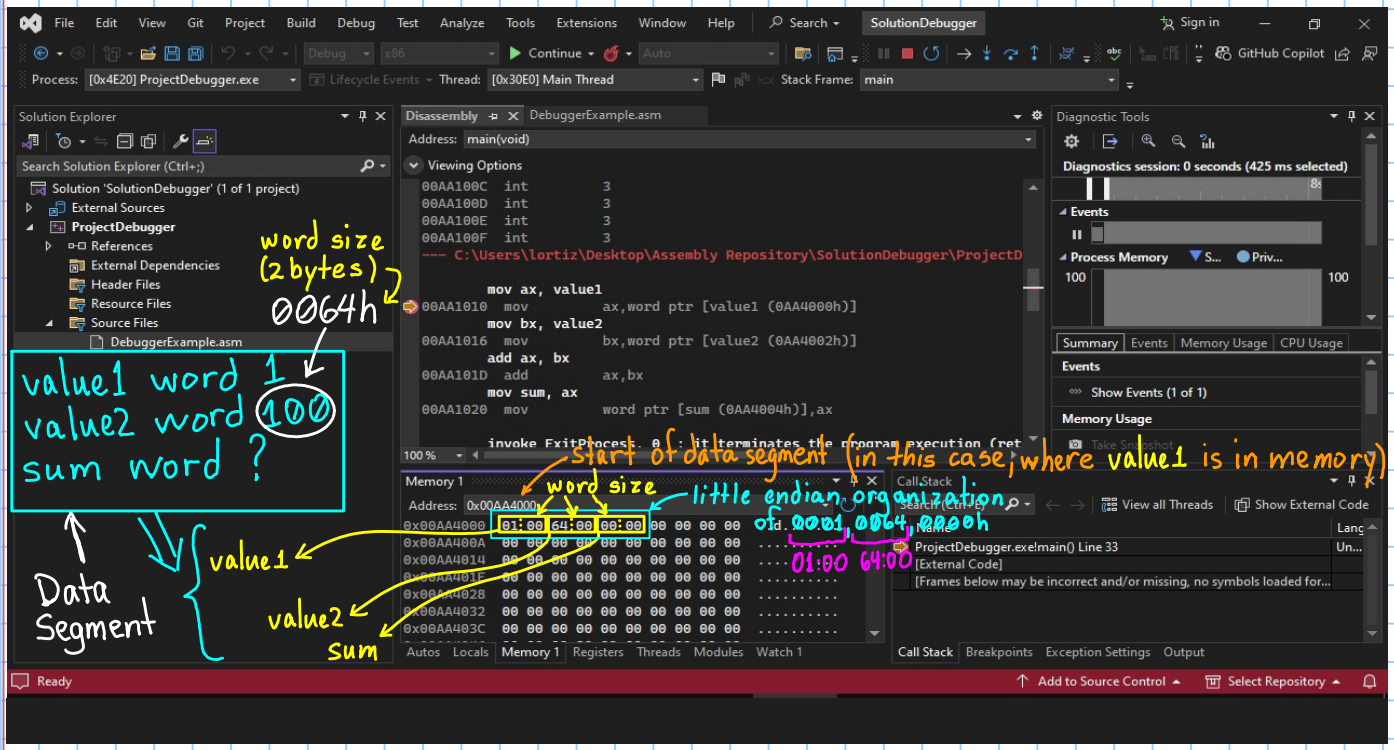
Show memory area assigned to program



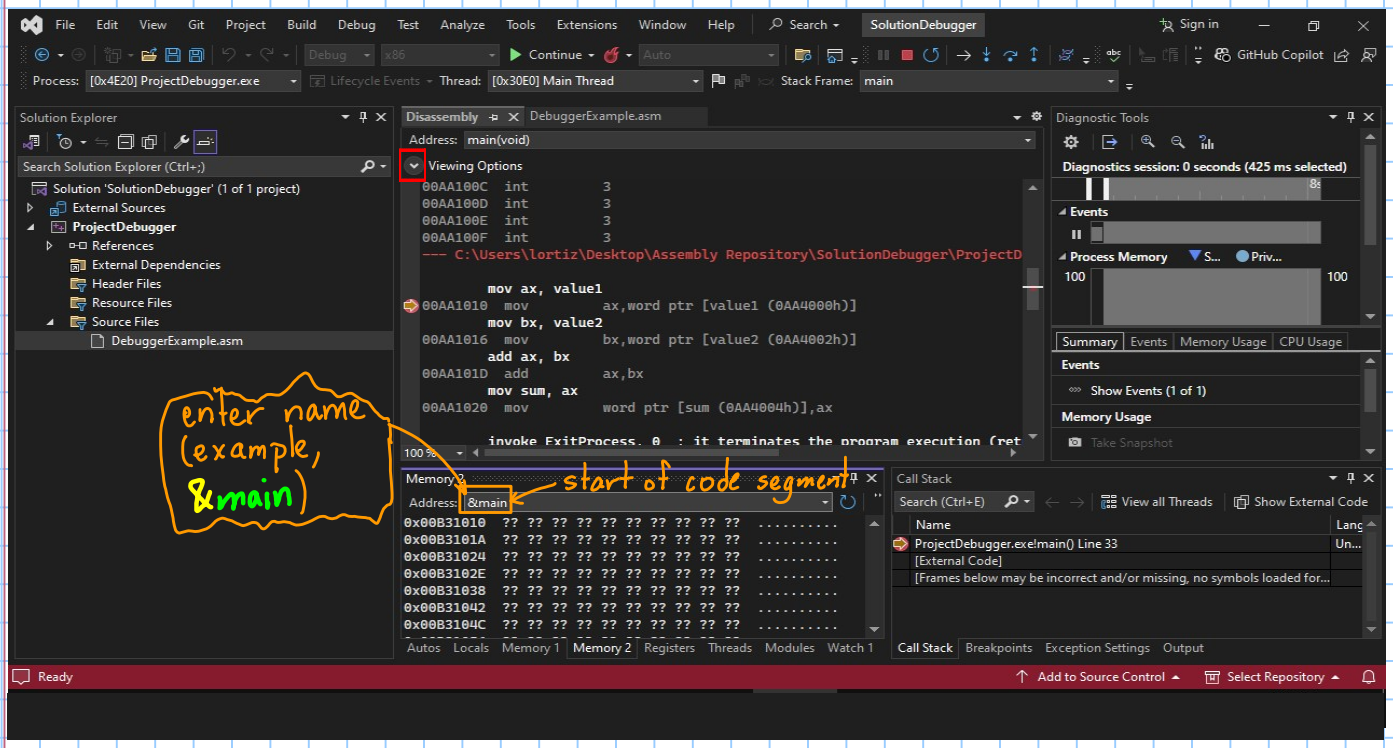


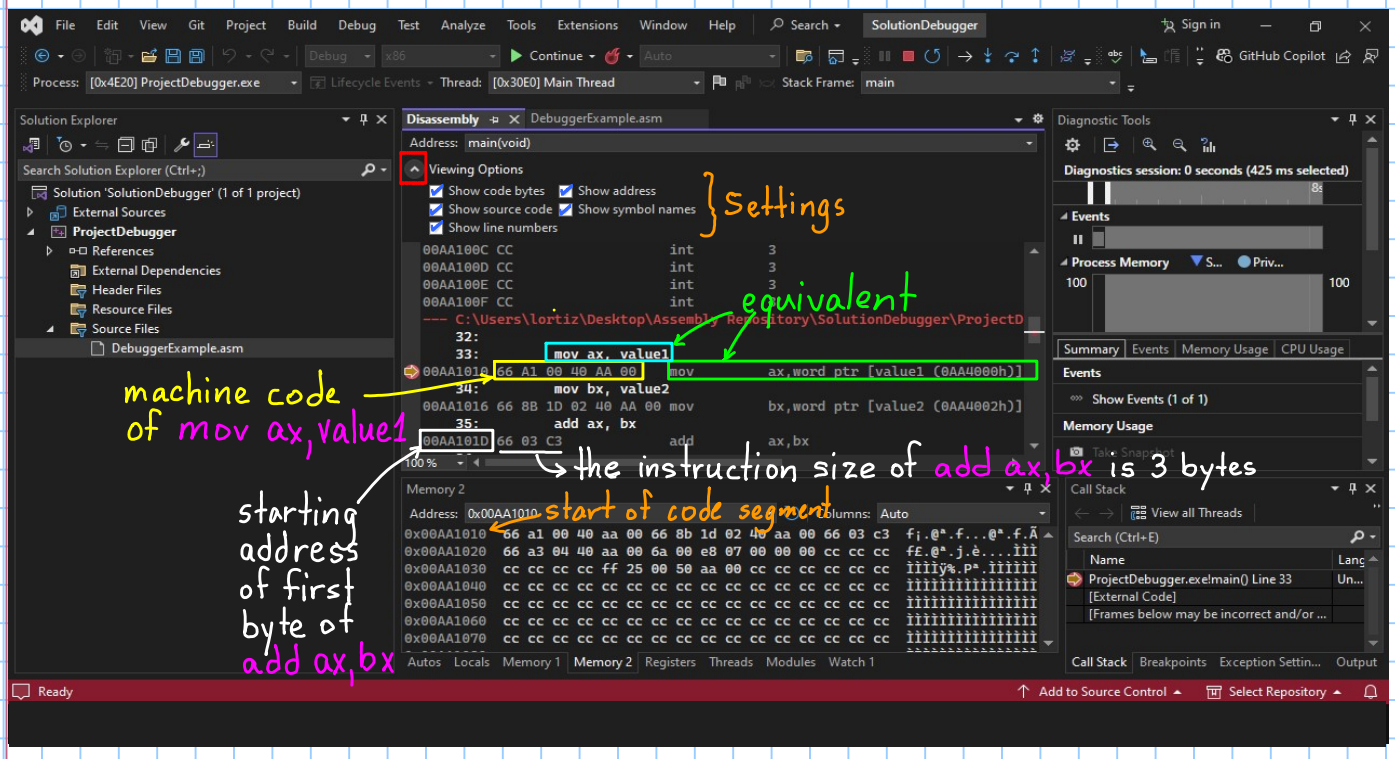
Show data segment

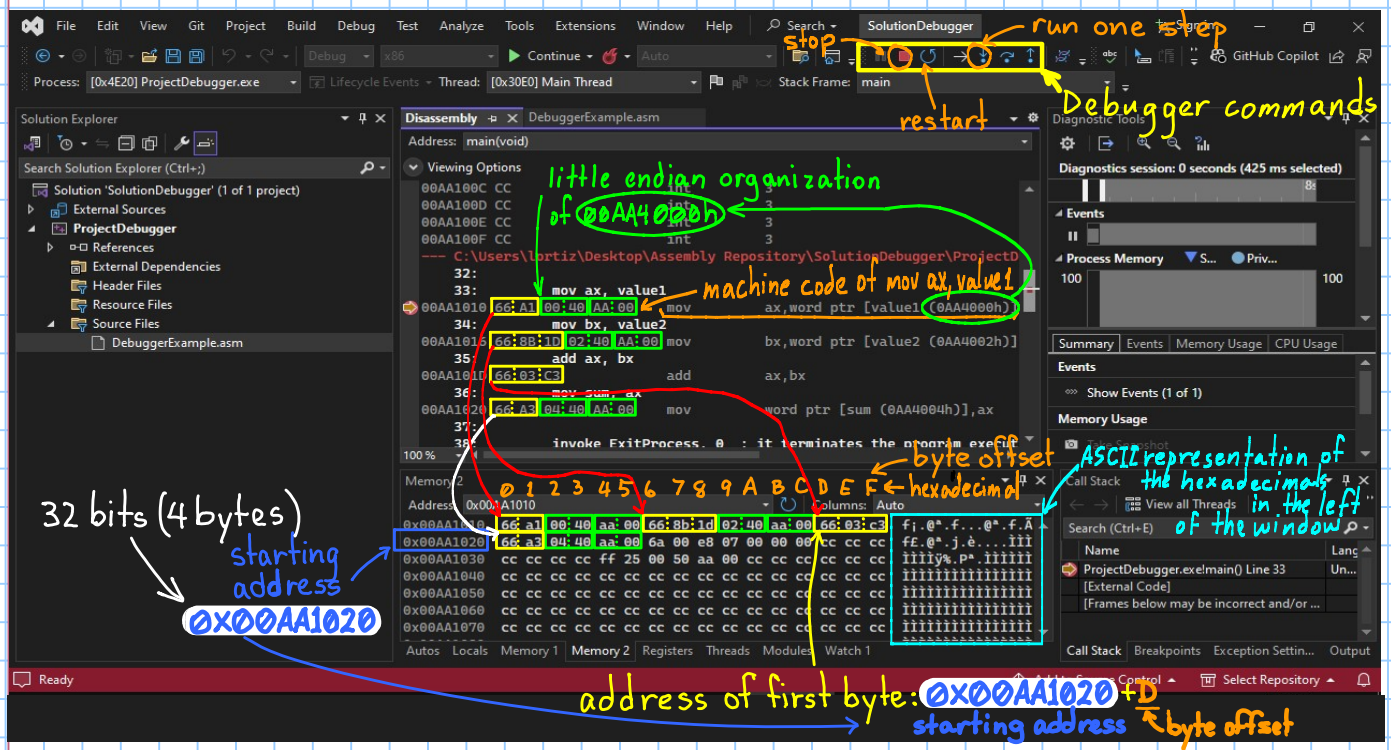


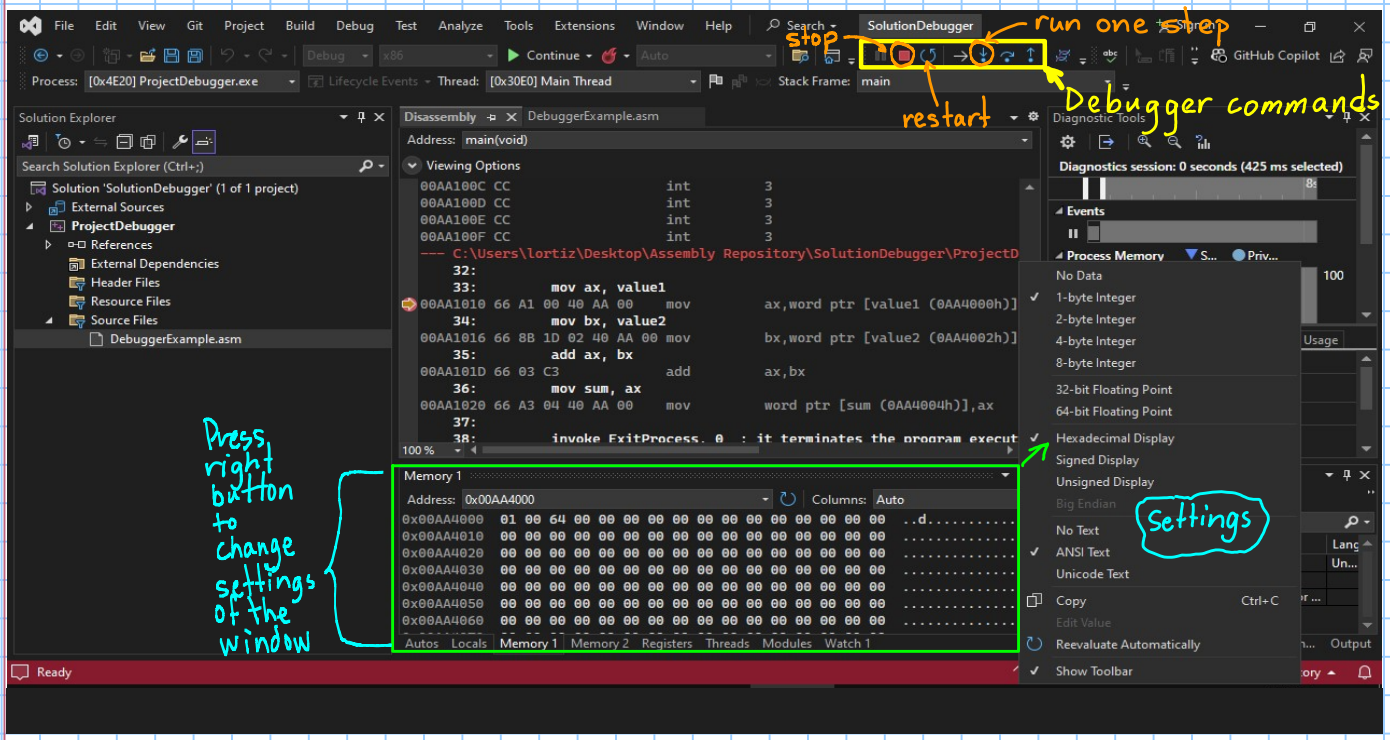


Show code segment






stop
run one step
restart
Debugger commands
little endian organization of 0x00AA4000h
machine code of mov ax, value1
byte offset
ASCII representation of the hexadecimals in the left of the window
0 1 2 3 4 5 6 7 8 9 A B C D E F ← hexadecimal
32 bits (4 bytes)
starting address
0x00AA1020
address of first byte: 0x00AA1020 + D
starting address
byte offset


stop
run one step
restart
Debugger commands
Press right button to change settings of the window
Settings
No Data
1-byte Integer
2-byte Integer
4-byte Integer
8-byte Integer
32-bit Floating Point
64-bit Floating Point
Hexadecimal Display
Signed Display
Unsigned Display
Big Endian
No Text
ANSI Text
Unicode Text
Copy Ctrl+C
Edit Value
Reevaluate Automatically
Show Toolbar

Running one step (one instruction)