

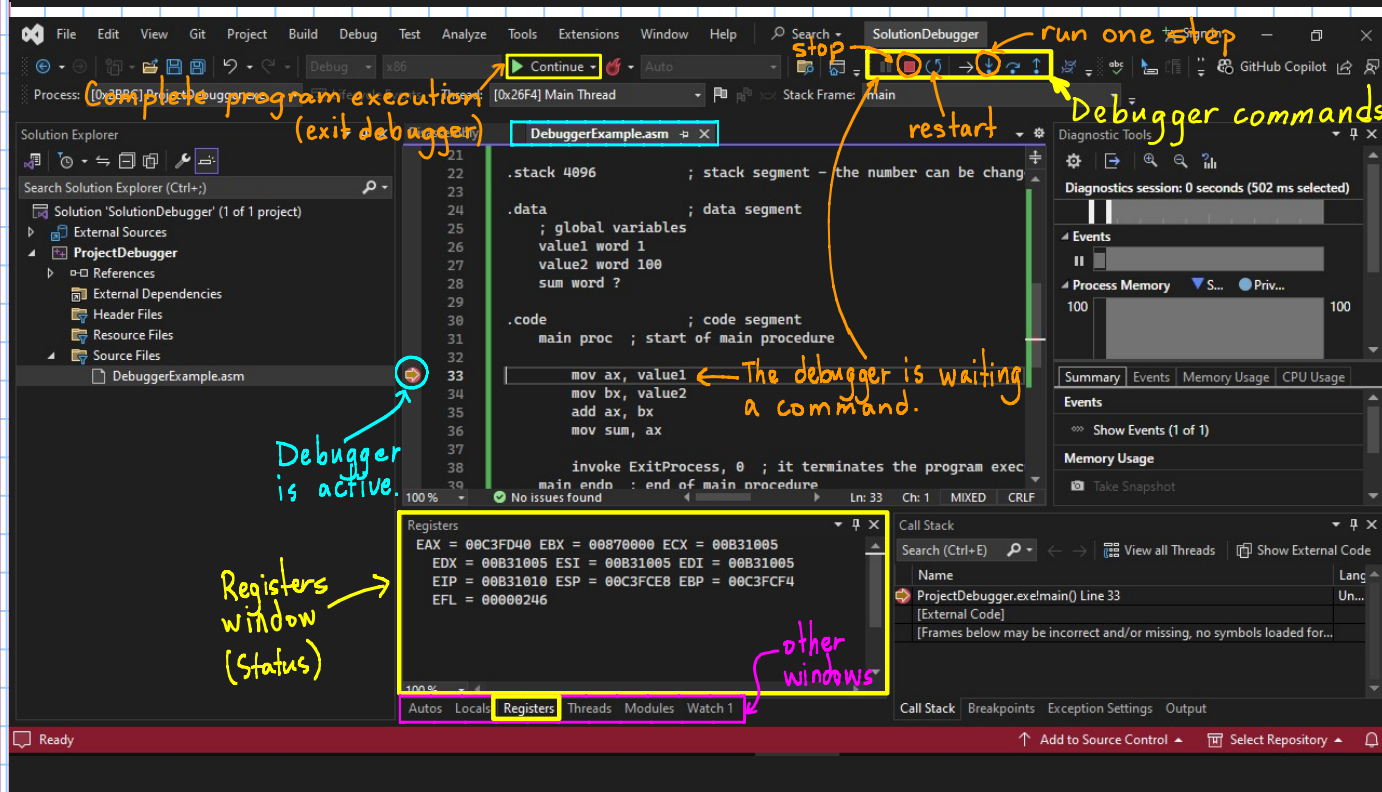
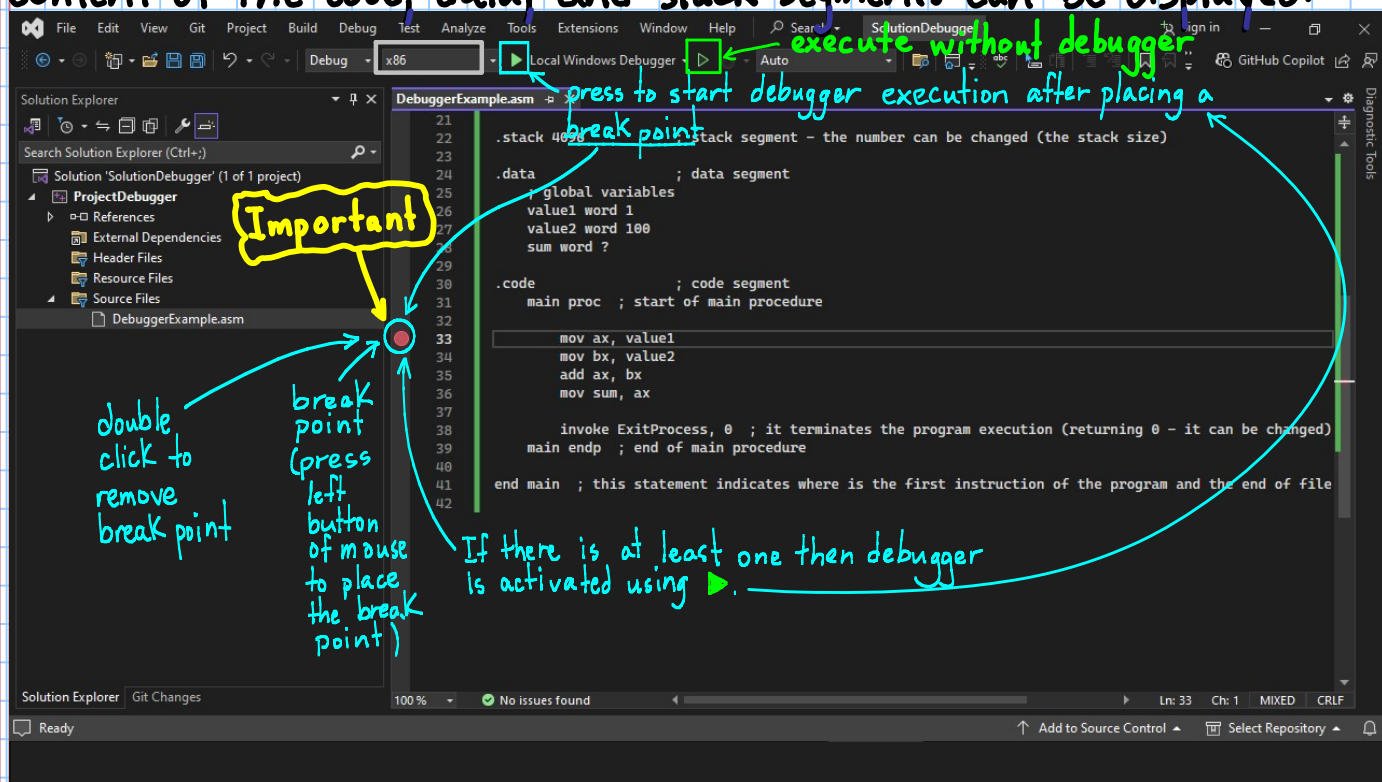
# How to use the debugger to run a program

Note Title

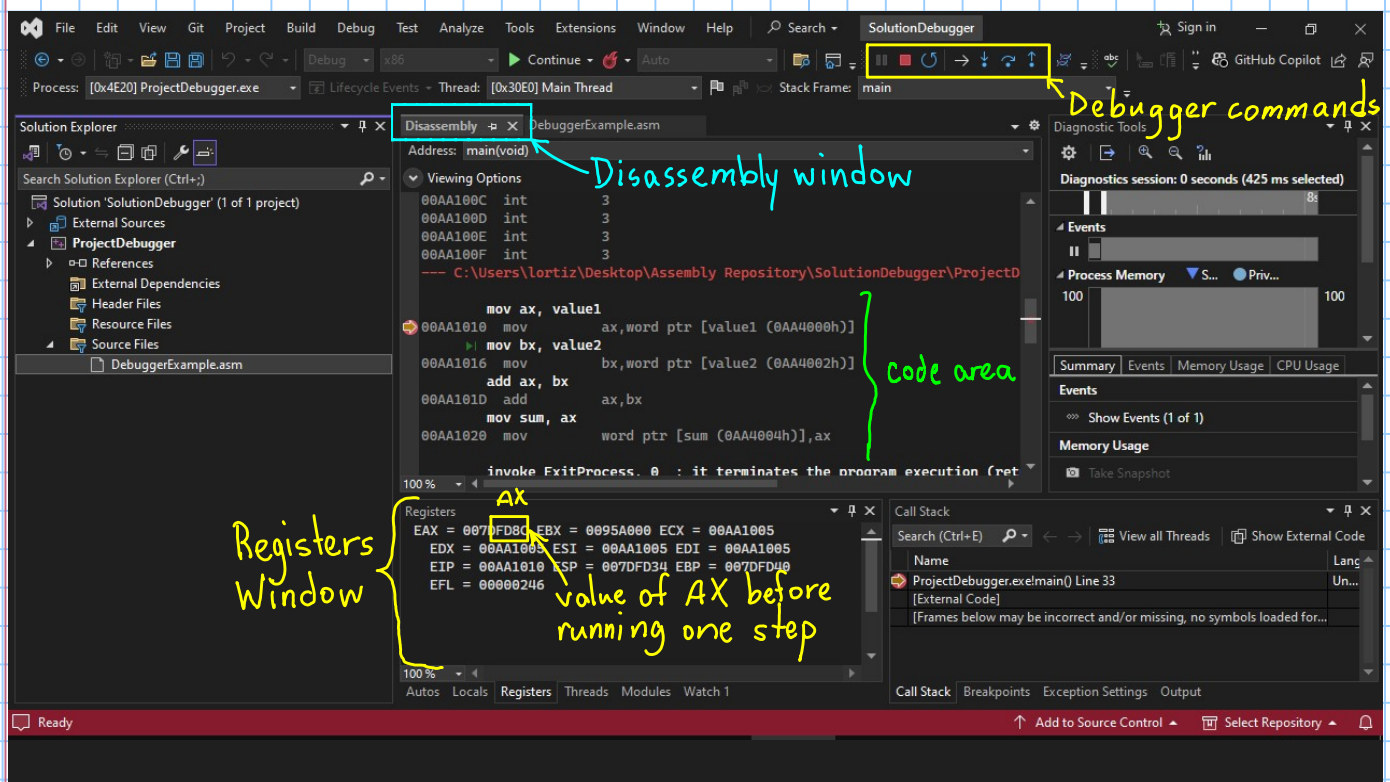
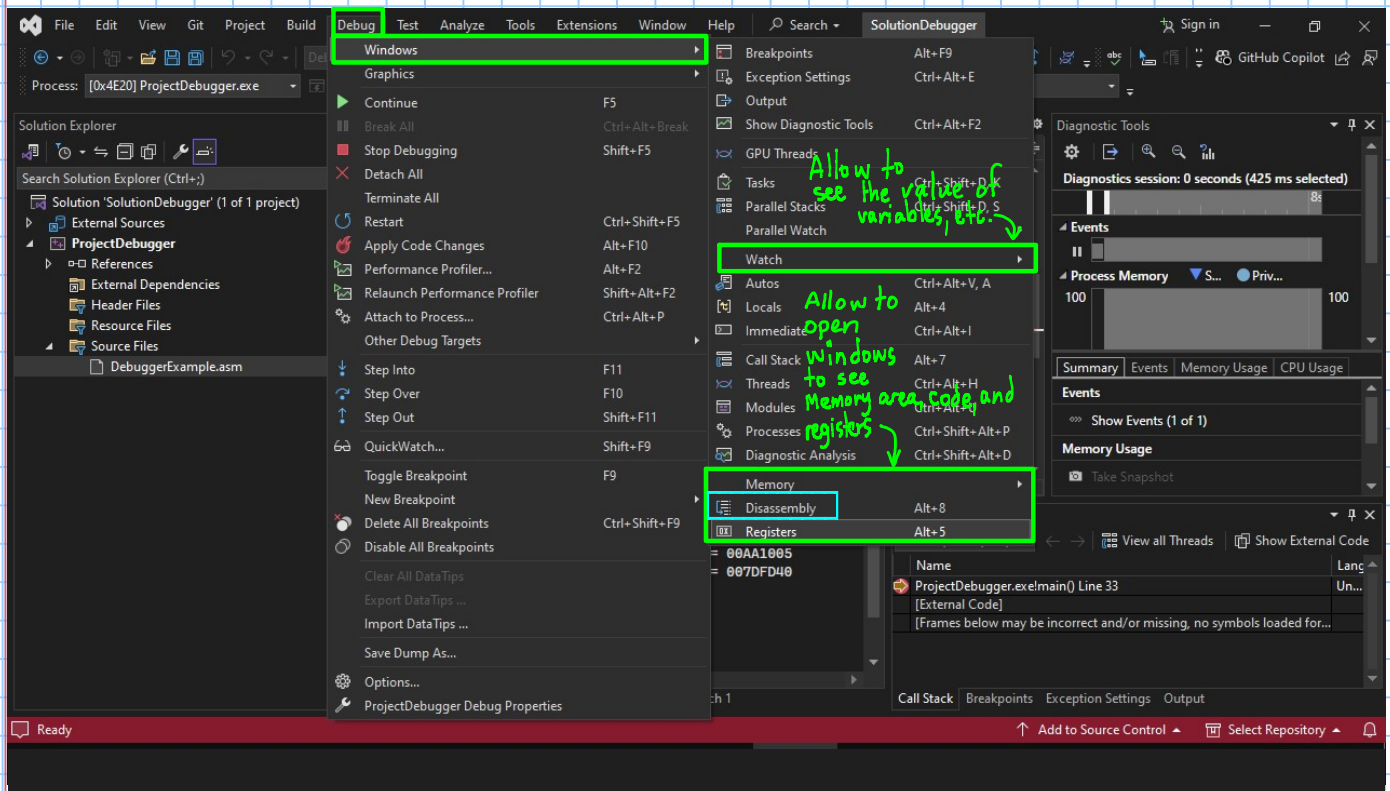
Start debugger execution (the images are for x86; similar for x64):

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.

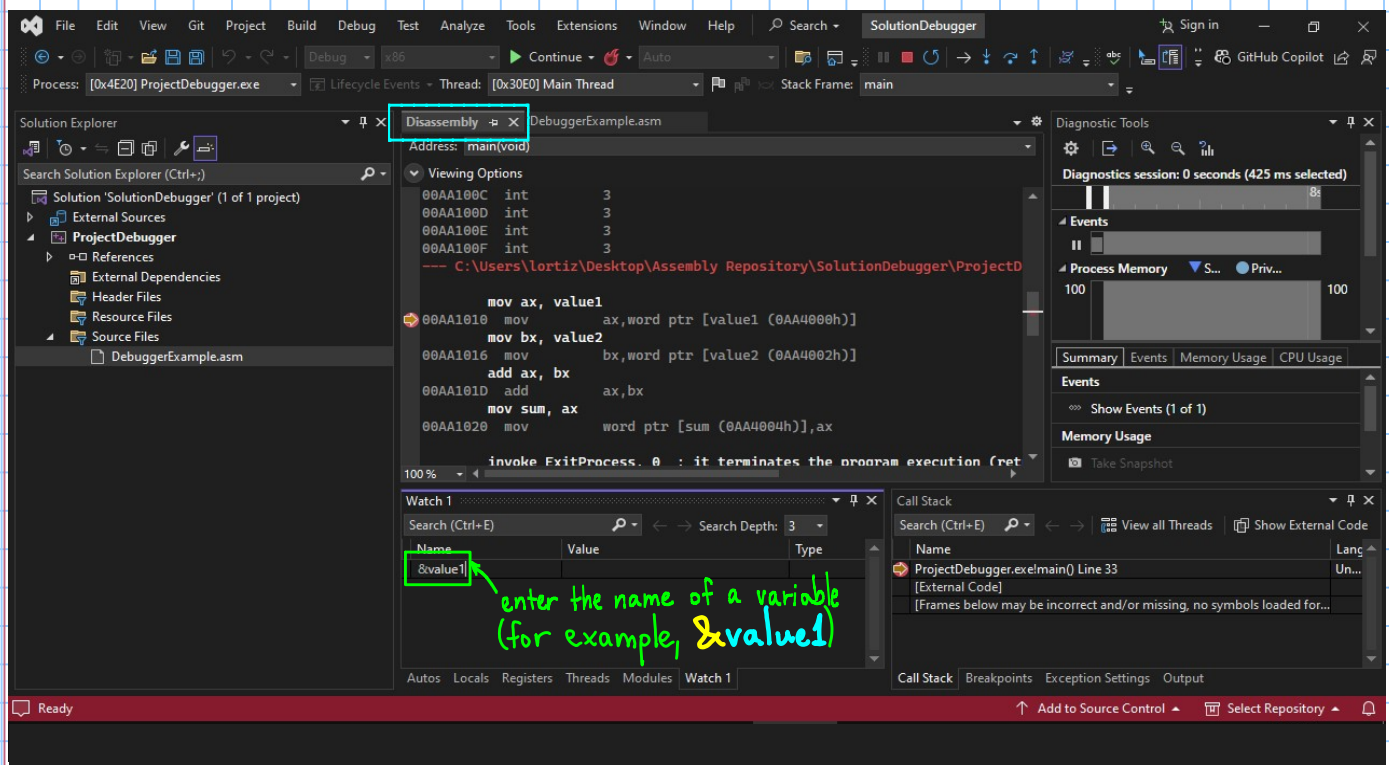
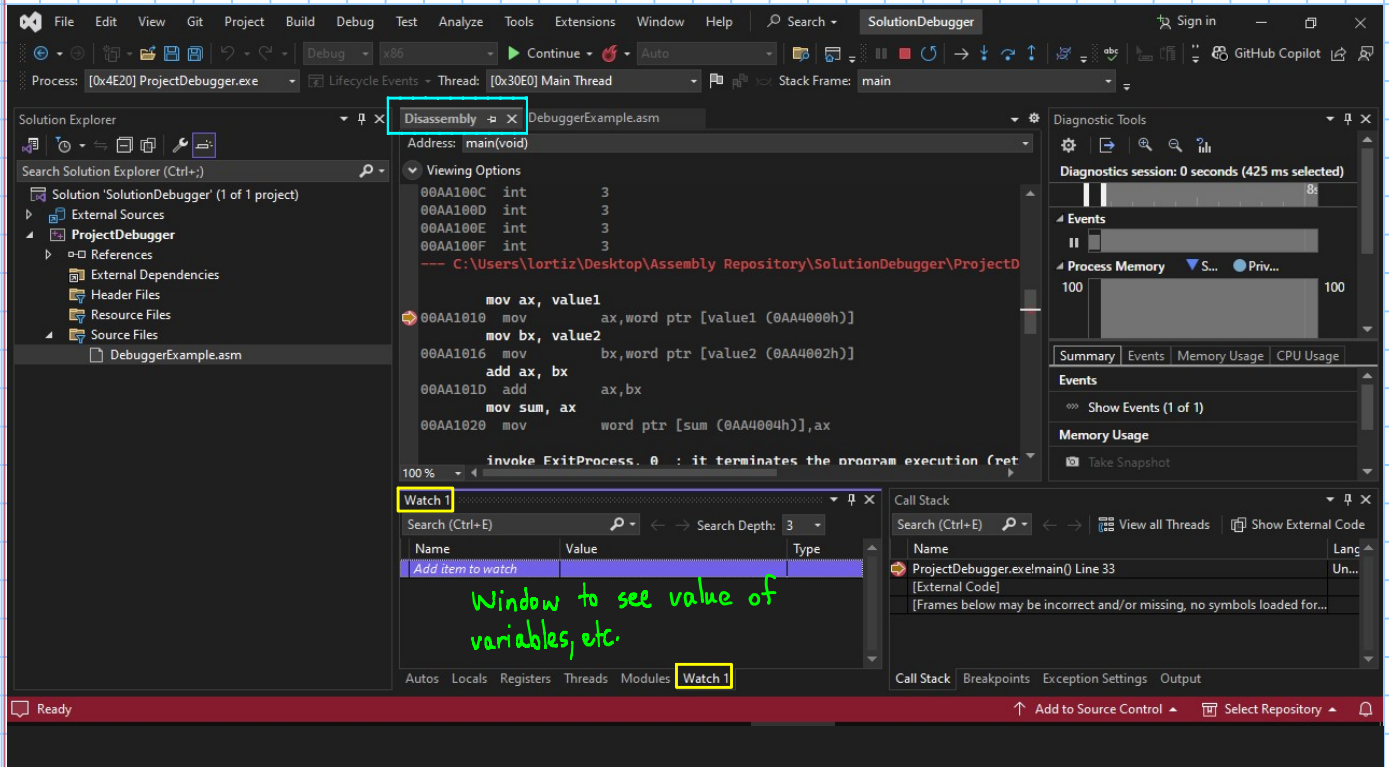


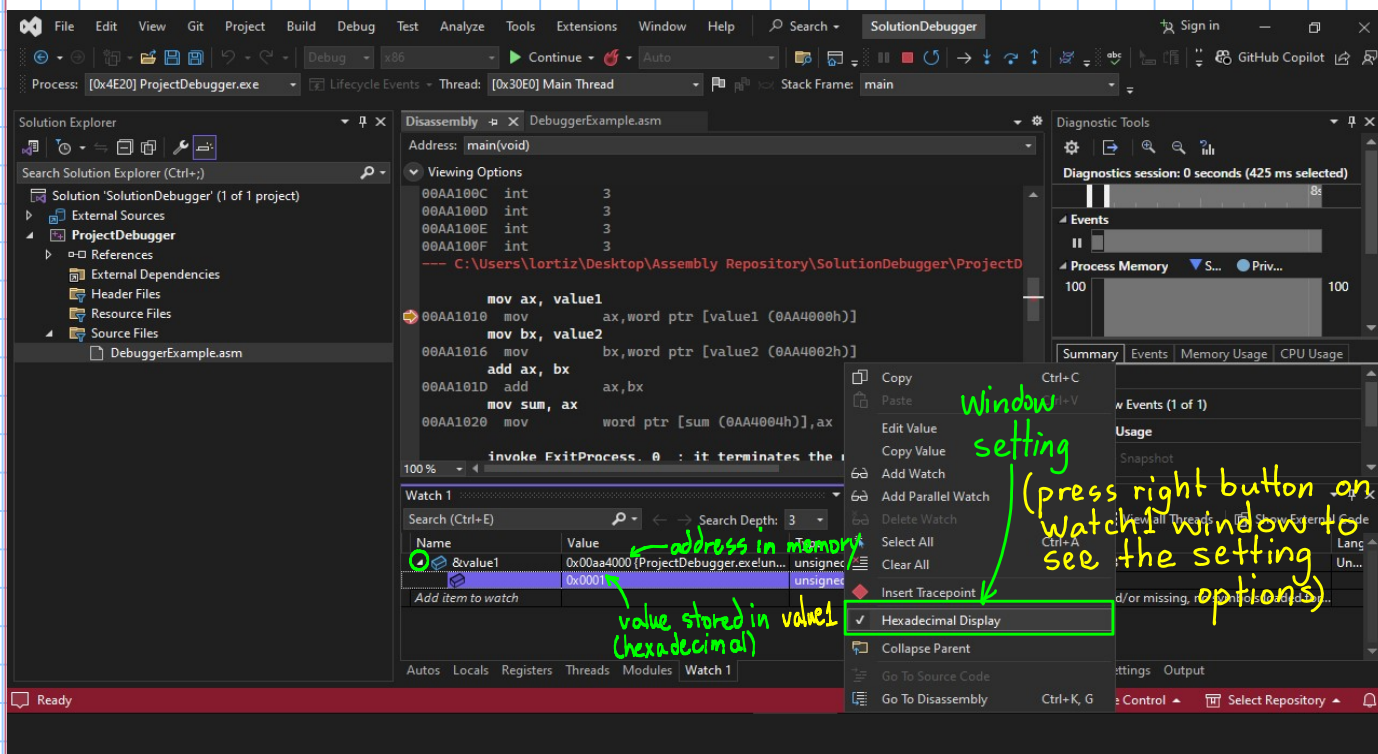
# Windows for showing memory area, registers, and machine code



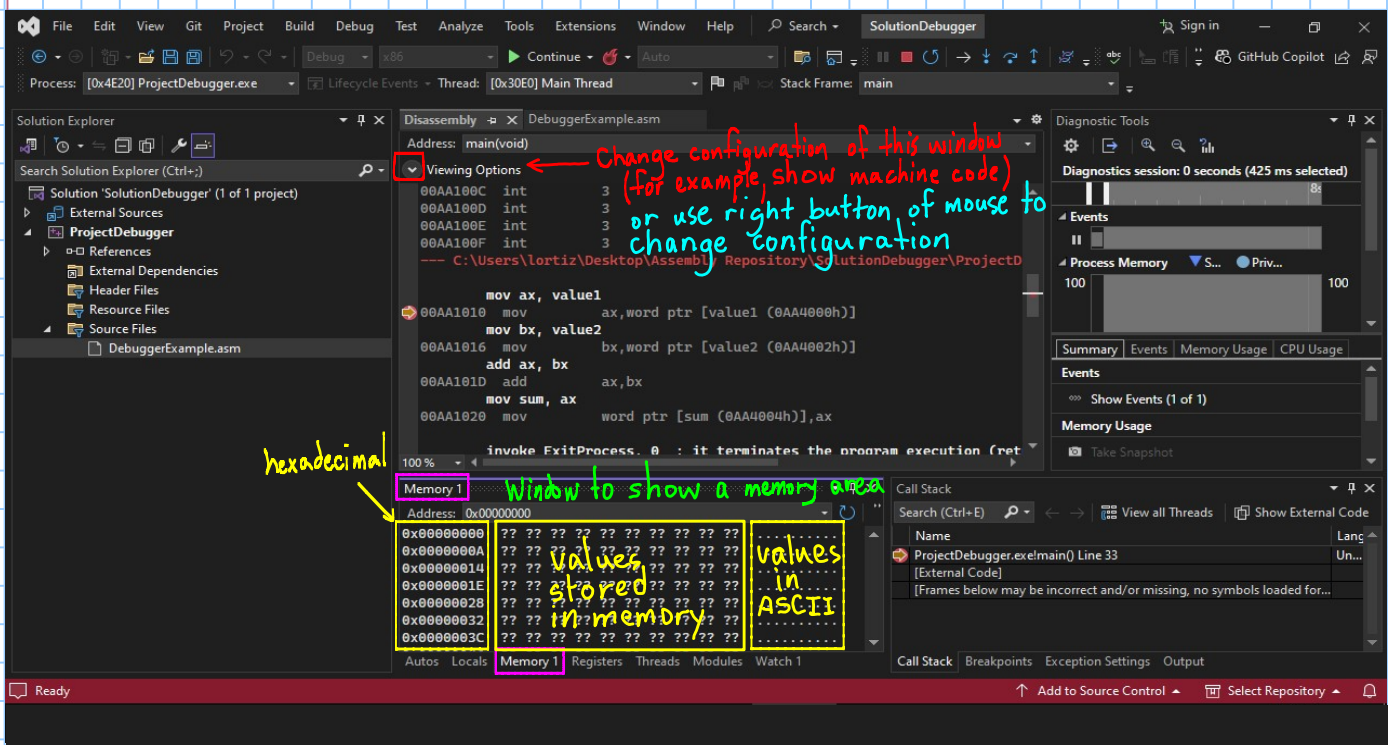
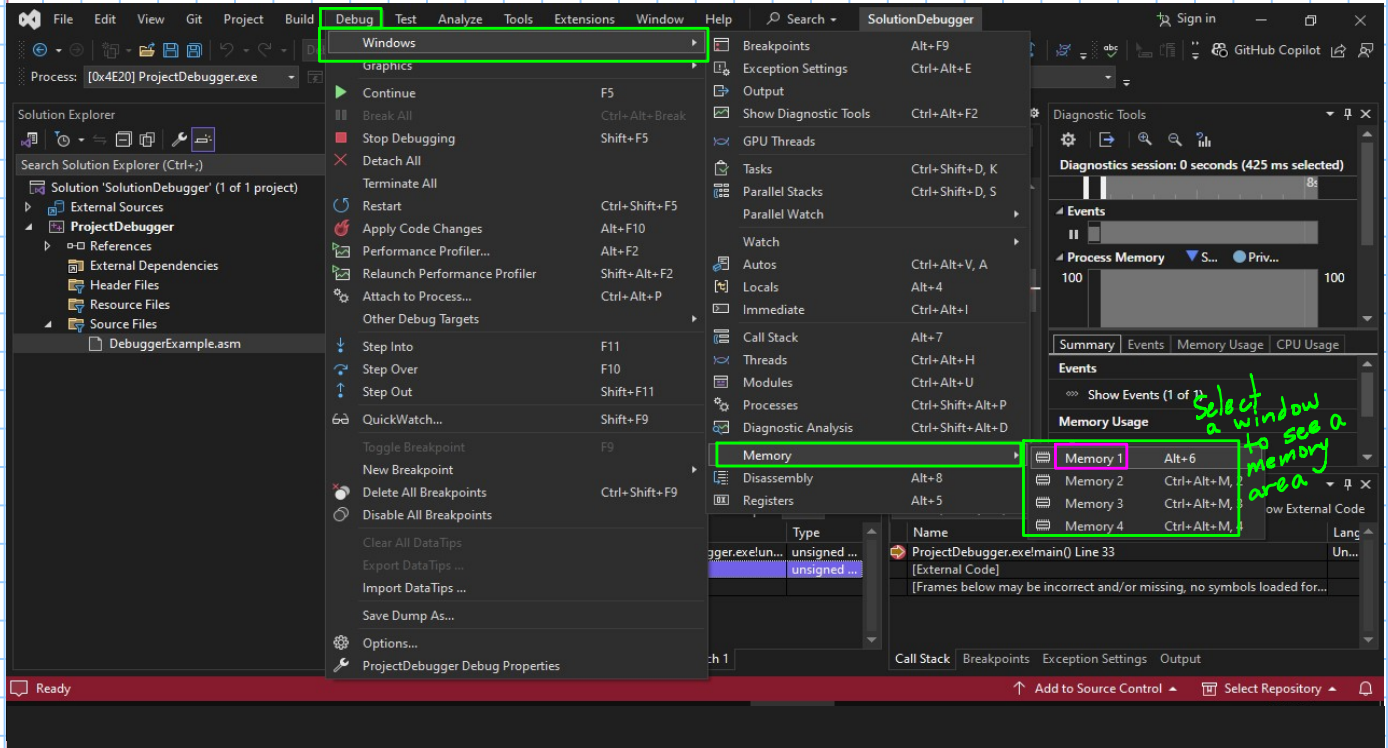


# Show global variables in data segment





# Show memory area assigned to program

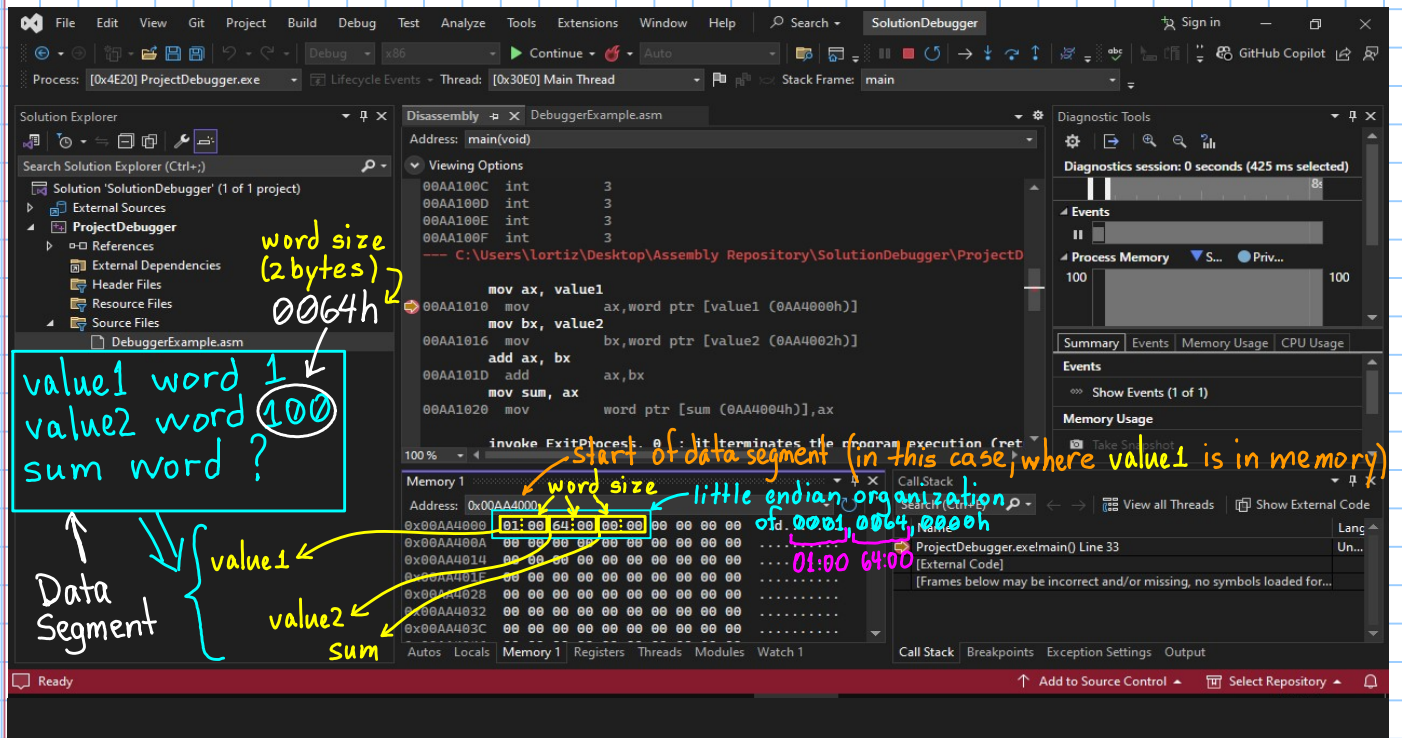




value1 word 1  
value2 word 100  
sum word?

Data Segment

start of data segment (in this case, where value1 is in memory)



# Show code segment

The screenshot shows the Visual Studio Solution Debugger interface. The Disassembly window displays assembly code for `DebuggerExample.asm`. A handwritten note in orange and green says "enter name (example, &main)" with an arrow pointing to the `main` symbol in the Memory window. Another handwritten note in orange says "start of code segment" with an arrow pointing to the `invoke ExitProcess. 0 : it terminates the program execution (ret` instruction.

Disassembly window (Address: main(void)):

```
00AA100C int 3
00AA100D int 3
00AA100E int 3
00AA100F int 3
---- C:\Users\lortiz\Desktop\Assembly Repository\SolutionDebugger\ProjectD
mov ax, value1
00AA1010 mov ax, word ptr [value1 (0AA4000h)]
mov bx, value2
00AA1016 mov bx, word ptr [value2 (0AA4002h)]
add ax, bx
00AA101D add ax, bx
mov sum, ax
00AA1020 mov word ptr [sum (0AA4004h)], ax
invoke ExitProcess. 0 : it terminates the program execution (ret
```

Memory window (Address: &main):

```
0x00B31010 ?? ?? ?? ?? ?? ?? ?? ??
0x00B3101A ?? ?? ?? ?? ?? ?? ?? ??
0x00B31024 ?? ?? ?? ?? ?? ?? ?? ??
0x00B3102E ?? ?? ?? ?? ?? ?? ?? ??
0x00B31038 ?? ?? ?? ?? ?? ?? ?? ??
0x00B31042 ?? ?? ?? ?? ?? ?? ?? ??
0x00B3104C ?? ?? ?? ?? ?? ?? ?? ??
```

The screenshot shows the Visual Studio Solution Debugger interface. The Disassembly window displays assembly code for `DebuggerExample.asm`. A handwritten note in orange says "Settings" with an arrow pointing to the "Viewing Options" section. Another handwritten note in green says "equivalent" with an arrow pointing to the `mov ax, value1` instruction. A handwritten note in yellow says "machine code of mov ax, value1" with an arrow pointing to the machine code `66 A1 00 40 AA 00`. A handwritten note in orange says "starting address of first byte of add ax, bx" with an arrow pointing to the address `0x00AA101D`. A handwritten note in orange says "the instruction size of add ax, bx is 3 bytes" with an arrow pointing to the instruction `add ax, bx`. A handwritten note in orange says "start of code segment" with an arrow pointing to the address `0x00AA1010`.

Disassembly window (Address: main(void)):

```
00AA100C CC int 3
00AA100D CC int 3
00AA100E CC int 3
00AA100F CC int 3
---- C:\Users\lortiz\Desktop\Assembly Repository\SolutionDebugger\ProjectD
32:
33: mov ax, value1
00AA1010 66 A1 00 40 AA 00 mov ax, word ptr [value1 (0AA4000h)]
34: mov bx, value2
00AA1016 66 8B 1D 02 40 AA 00 mov bx, word ptr [value2 (0AA4002h)]
35: add ax, bx
00AA101D 66 03 C3 add ax, bx
```

Memory window (Address: 0x00AA1010):

```
0x00AA1010 66 A1 00 40 AA 00 66 8B 1D 02 40 AA 00 66 03 C3 f1.0*.f...0*.f.A
0x00AA1020 66 A3 04 40 AA 00 6A 00 E8 07 00 00 00 CC CC ff.0*.j.e....iii
0x00AA1030 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC iiii%.p%.iiii
0x00AA1040 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
0x00AA1050 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
0x00AA1060 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
0x00AA1070 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
```







# Running one step (one instruction)

