# Comp 3350: Computer Organization & Assembly Language

# HW # 5: Theme: Data Definitions, Addressing Modes, Arrays

*All main questions carry equal weight.*

*(Credit awarded to only those answers for which work has been shown.)*

1. [Memory Map] Fill in the following memory diagram with the data provided below. Please assume that the data segment begins at 0x0040400A.

.data

Alpha WORD 0CDh, 1234h

Beta BYTE 56h

Gamma DWORD 01234ABCDh

Delta BYTE 77h

|  |  |  |
| --- | --- | --- |
| Address | Variable | Data |
| 0040400A | Alpha | CD |
| 0040400B | +1 | 00 |
| 0040400C | +2 | 34 |
| 0040400D | +3 | 12 |
| 0040400E | Beta | 56 |
| 0040400F | Gamma | CD |
| 00404010 | +1 | AB |
| 00404011 | +2 | 34 |
| 00404012 | +3 | 12 |
| 00404013 | Delta | 77 |

1. [Addressing Modes] Copy the following code into your assembly development environment and single-step through it. For each single step execution, submit the screenshot. For those instructions referencing memory, do the linear address computation (typewritten/handwritten).

TITLE Addressing Modes (main.asm)

INCLUDE Irvine32.inc

.data

alpha DWORD 10203040h, 50607080h

beta DWORD 90A0B0C0h, D0E0F000h

gamma DWORD 1234h

.code

main PROC

mov eax, ABCDh; Immediate

mov ecx, eax; Register to Register

mov edi, OFFSET beta; Immediate

mov [gamma], eax; Indirect

mov esi, [gamma]; Direct -

mov esi, 4; Immediate

mov eax, beta[esi]; Indirect-offset

mov ebx, OFFSET alpha; Immediate

mov eax, [ebx]; Indirect

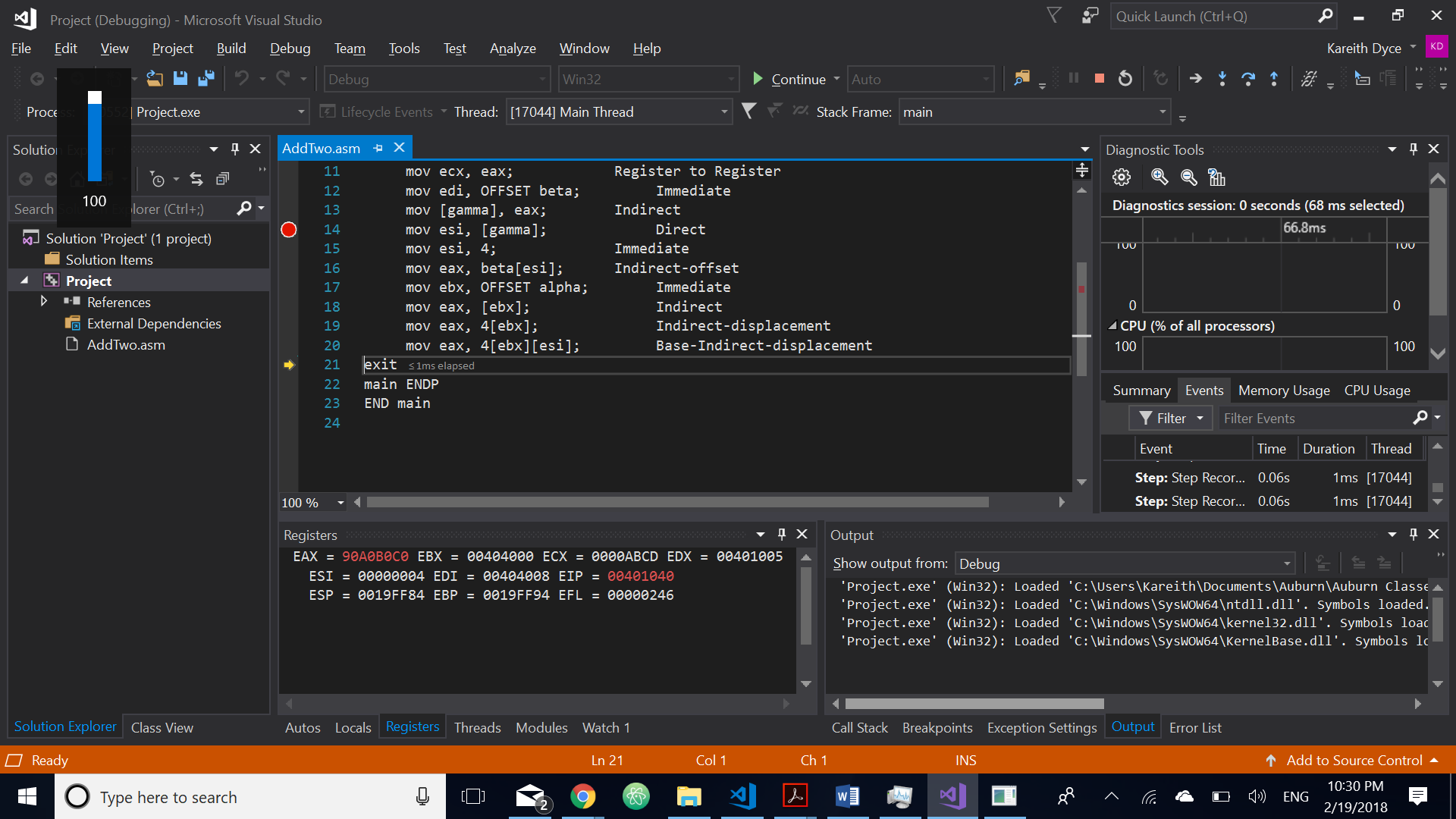
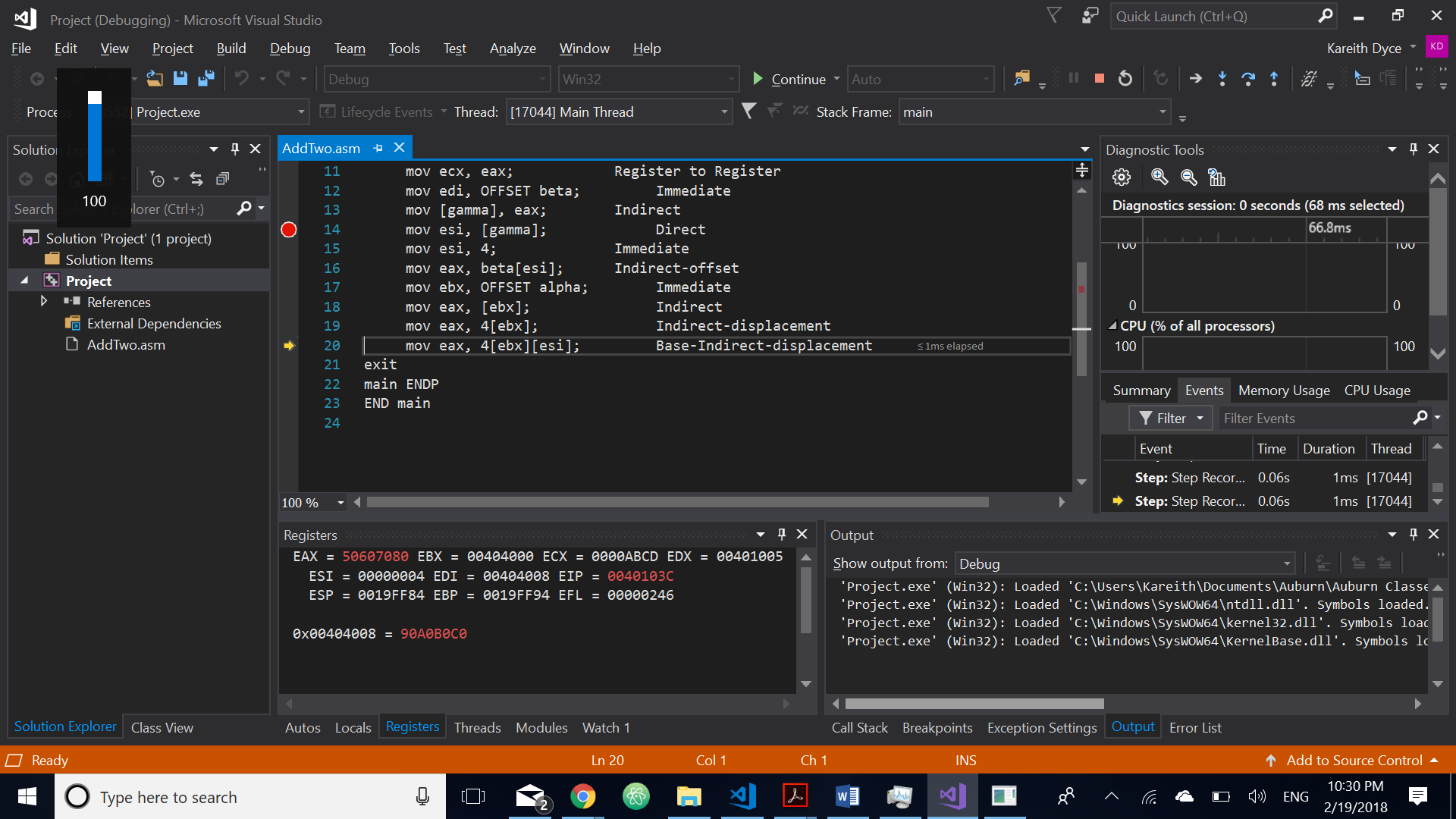
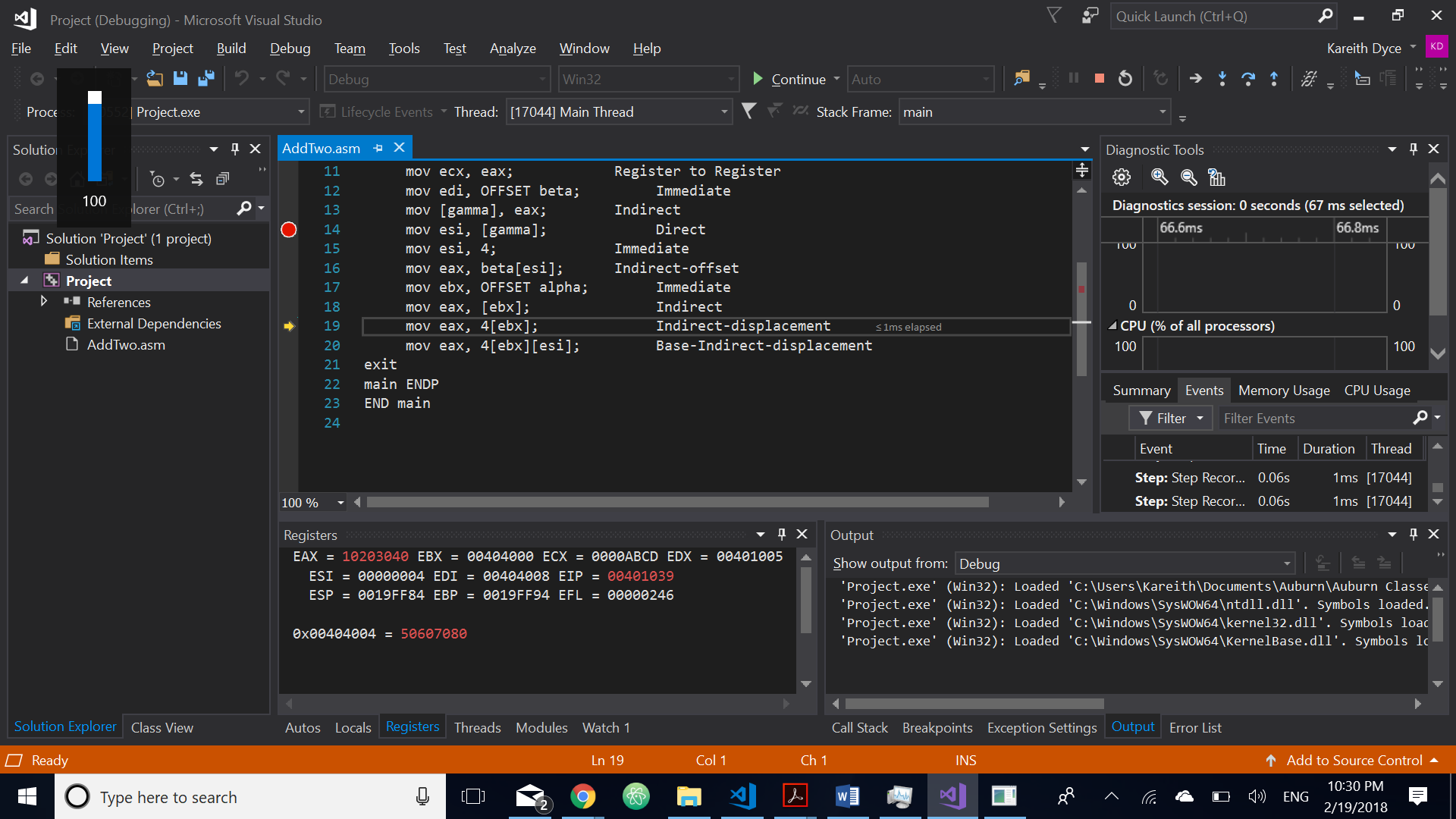
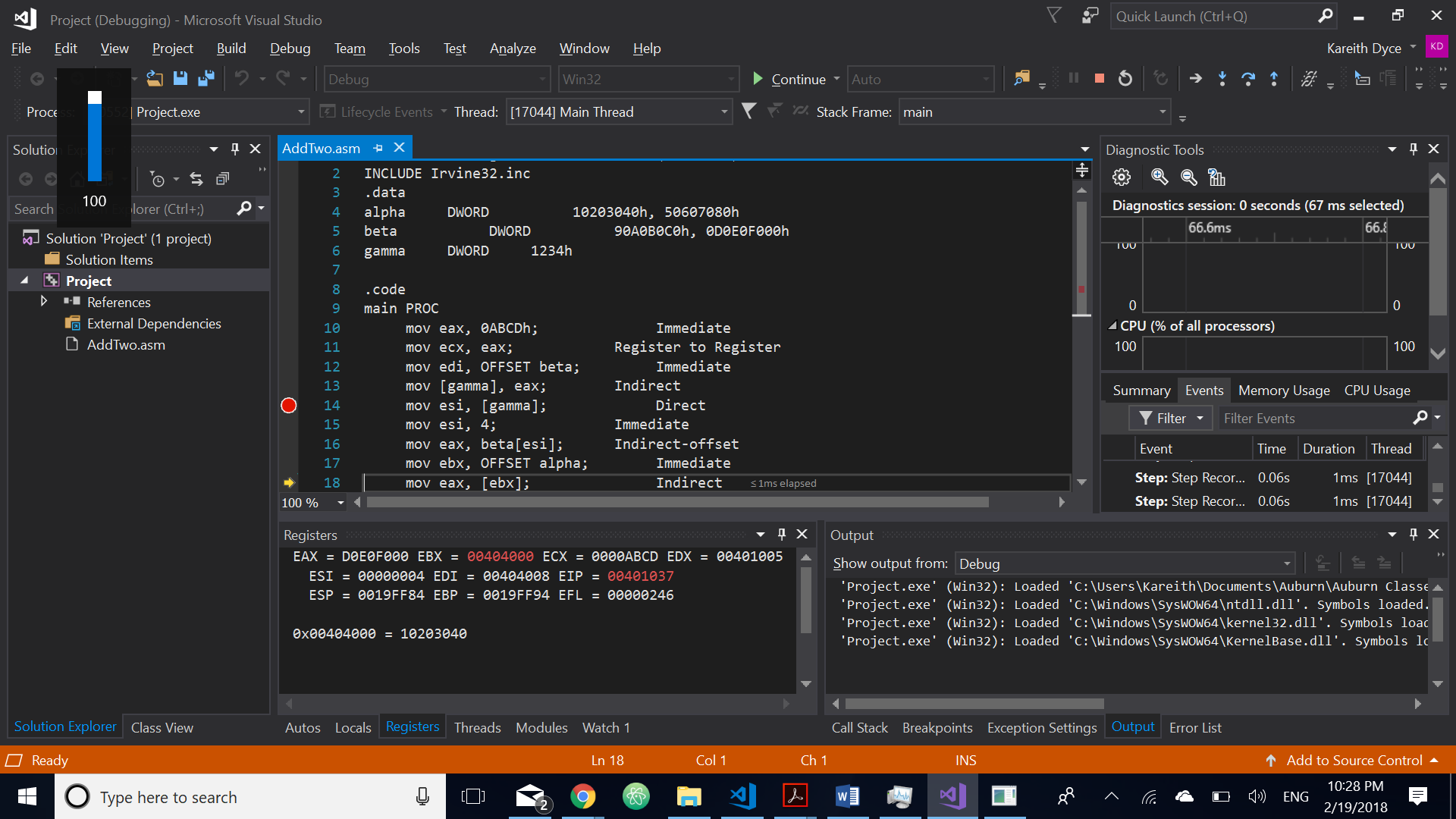
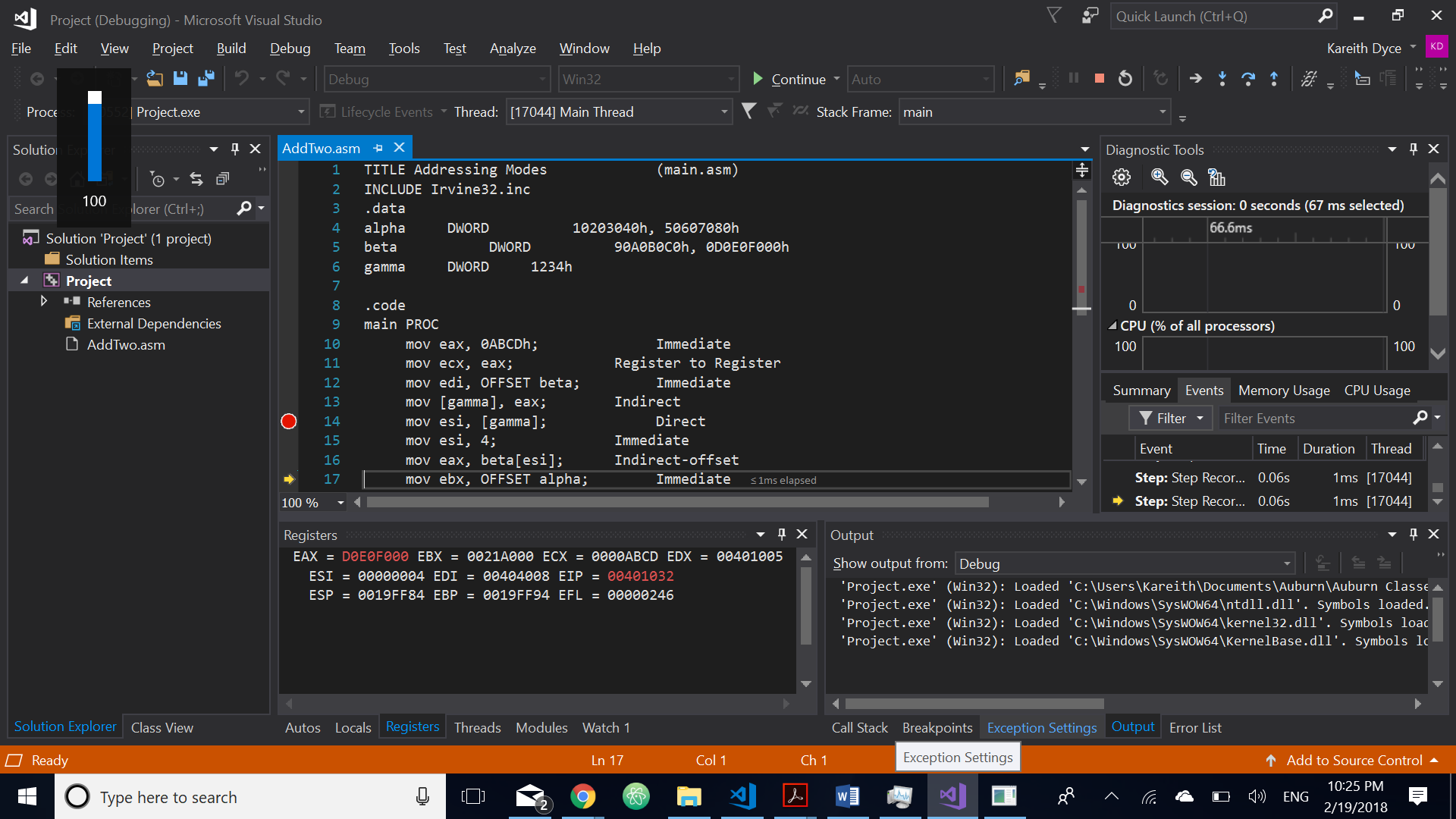
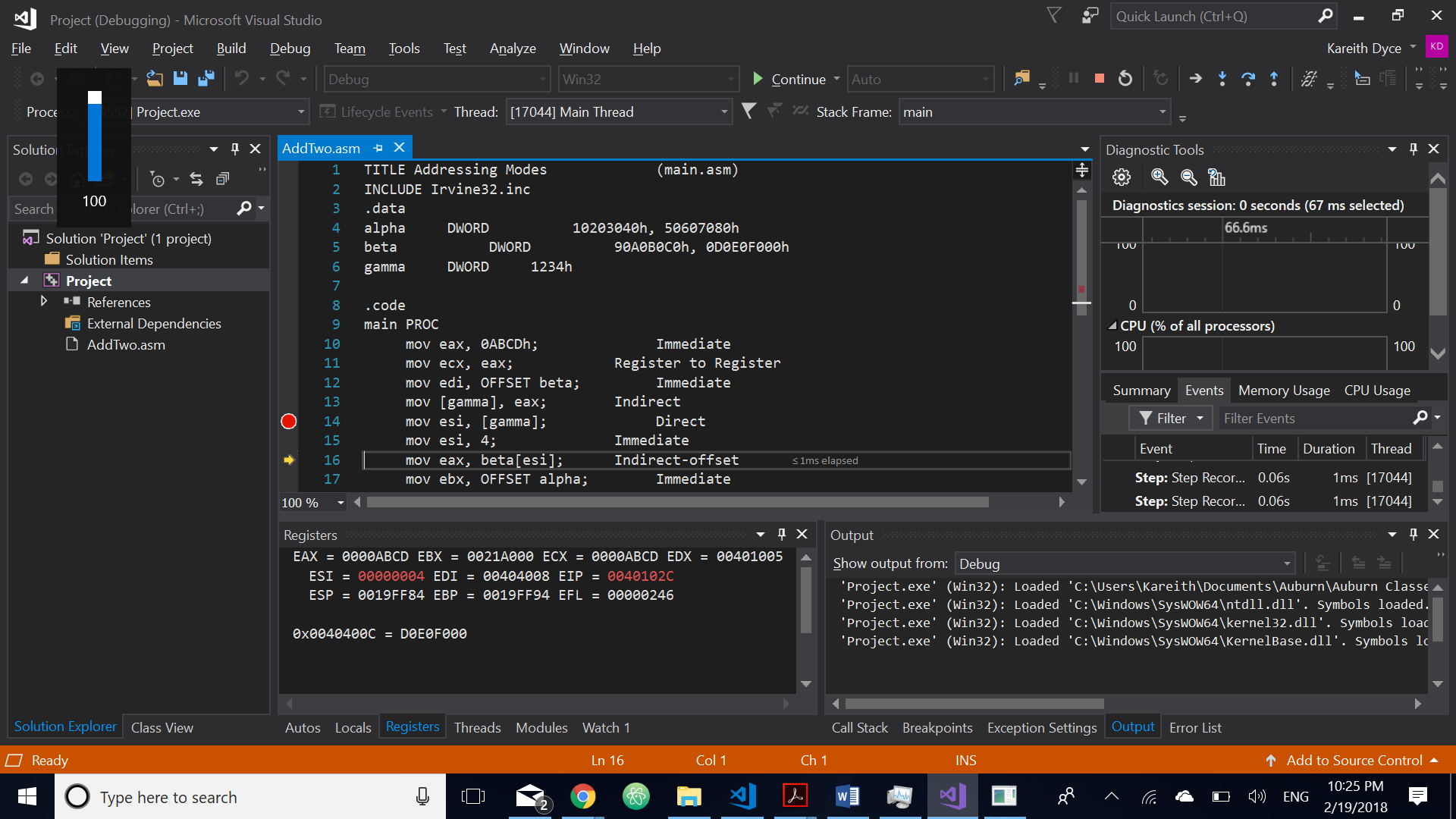
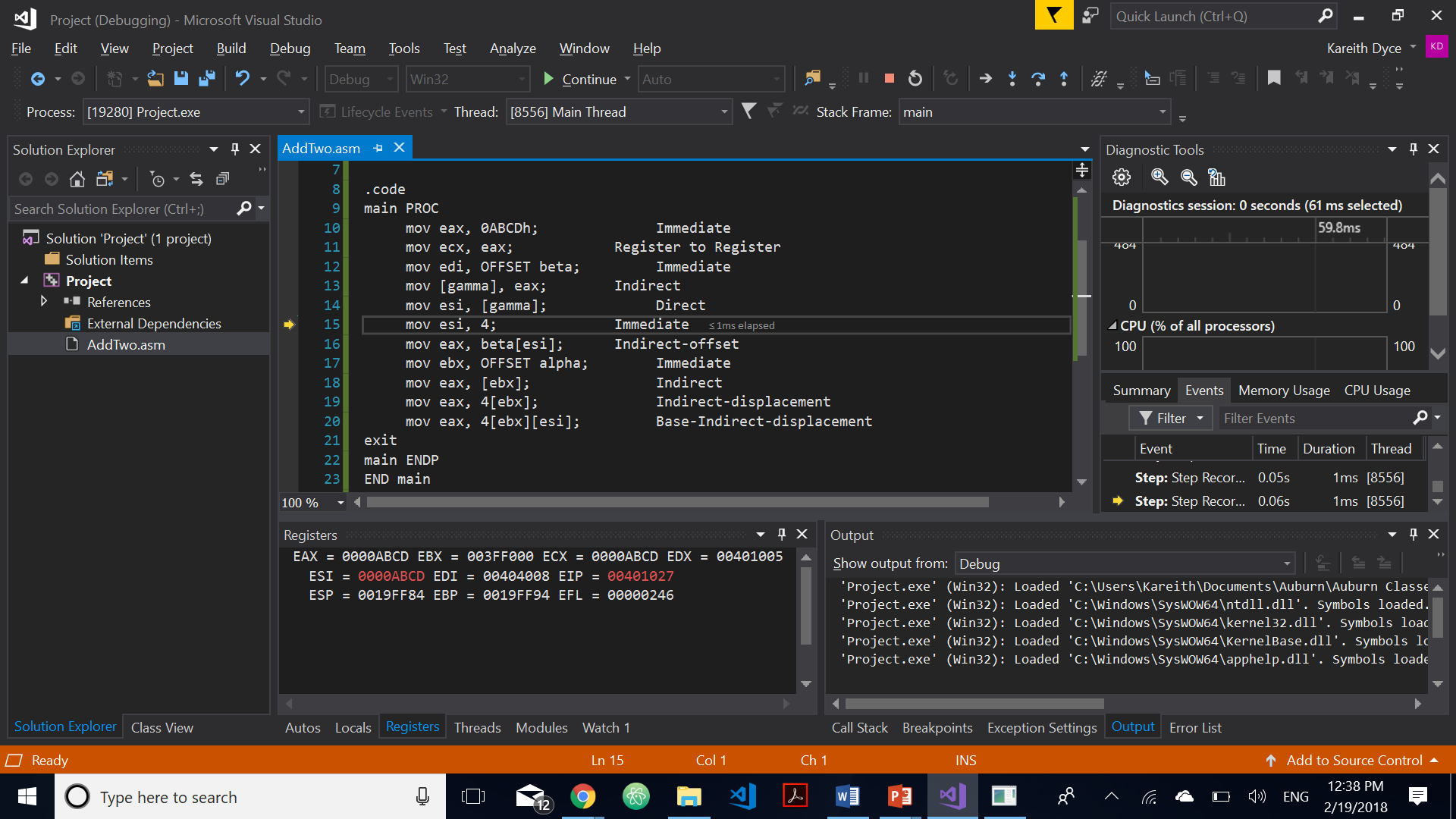
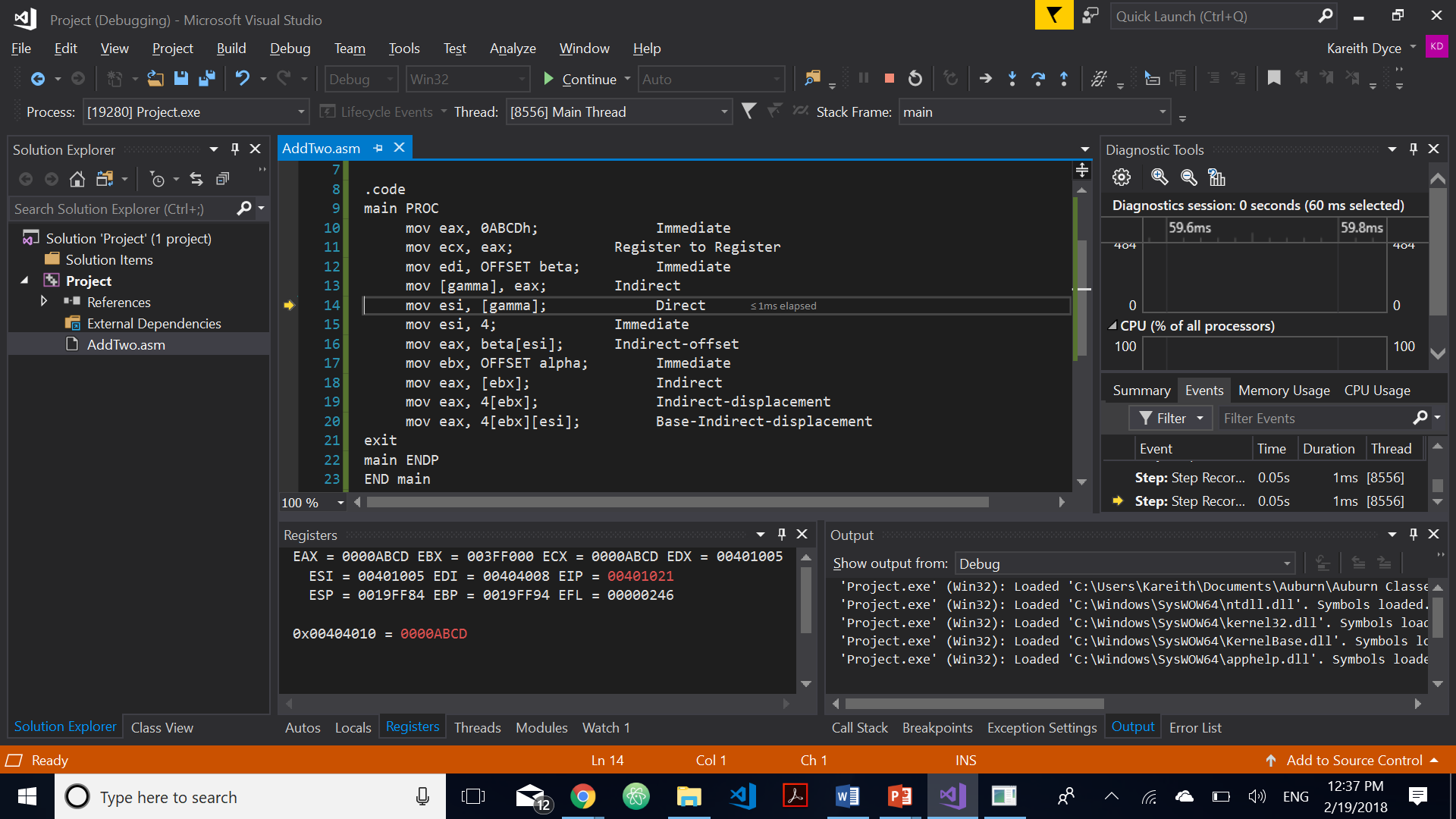
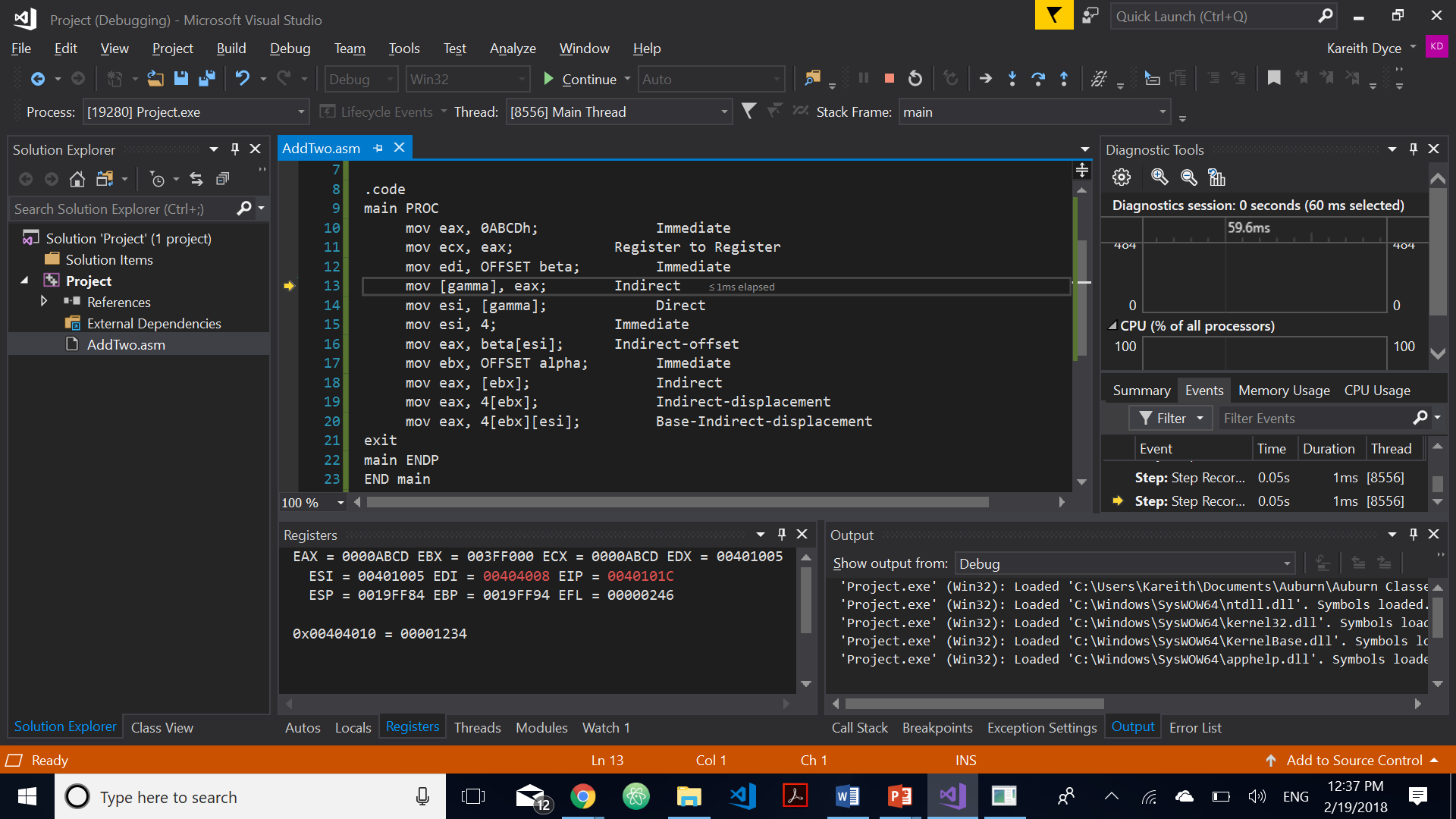
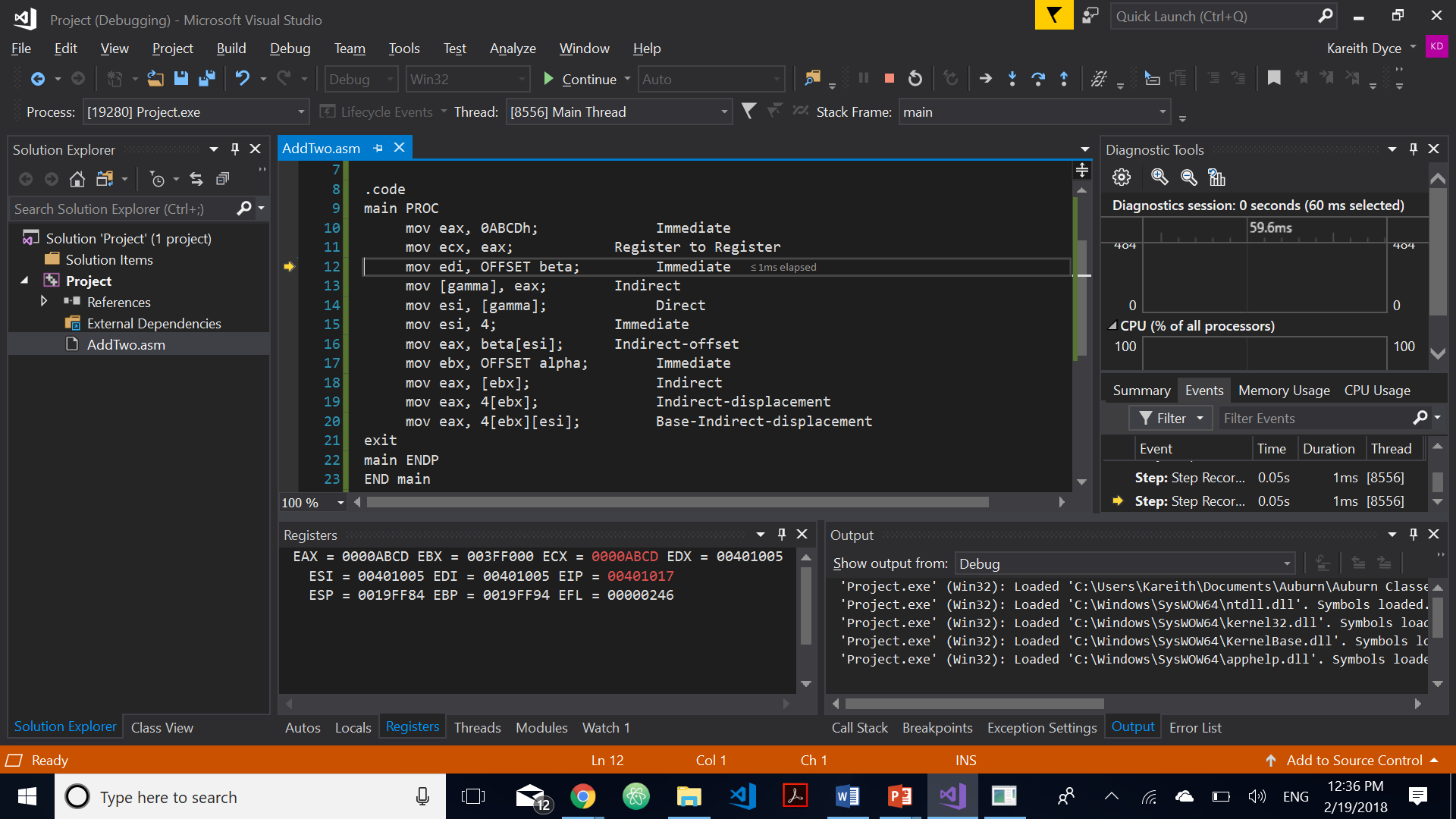
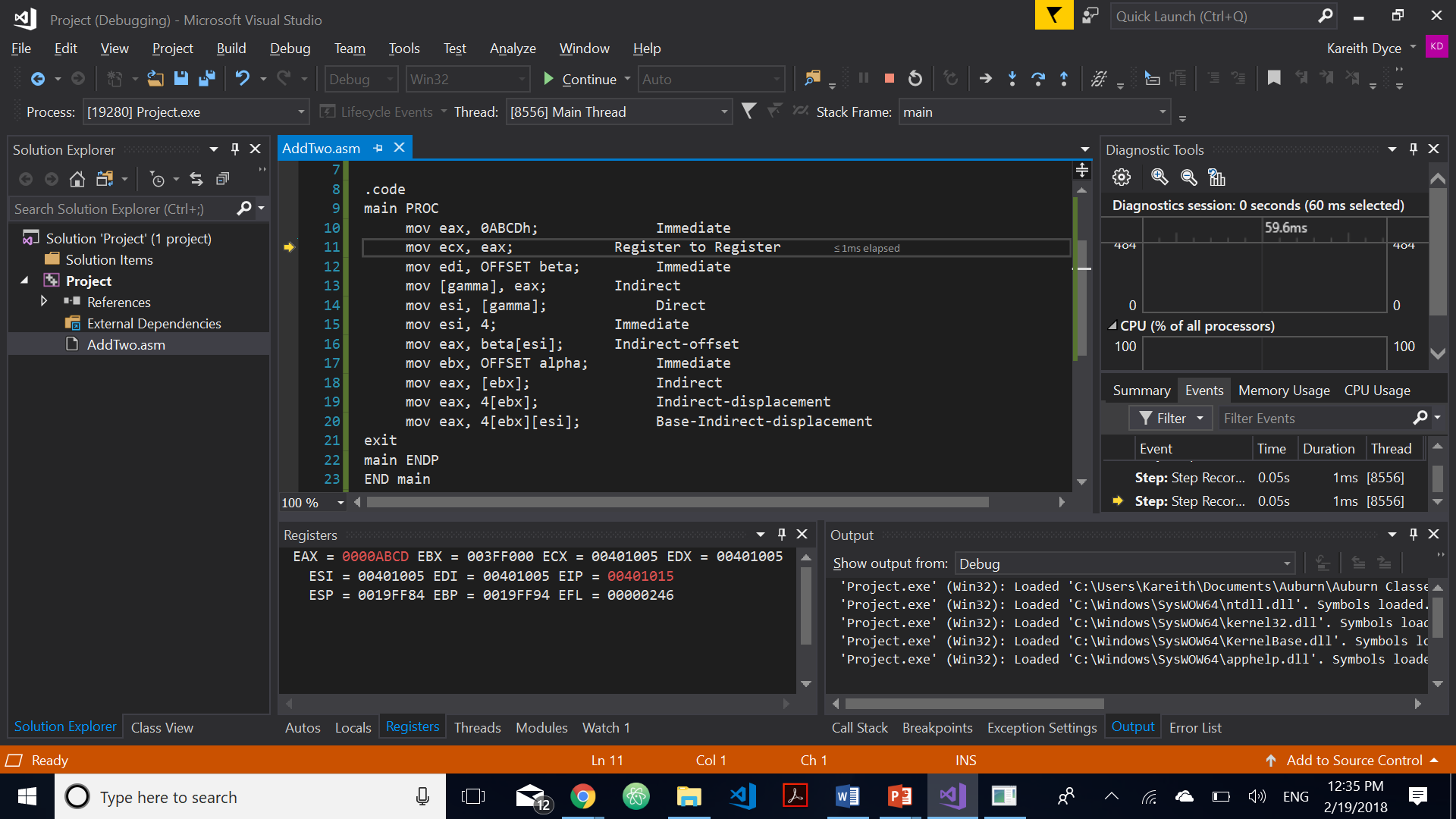
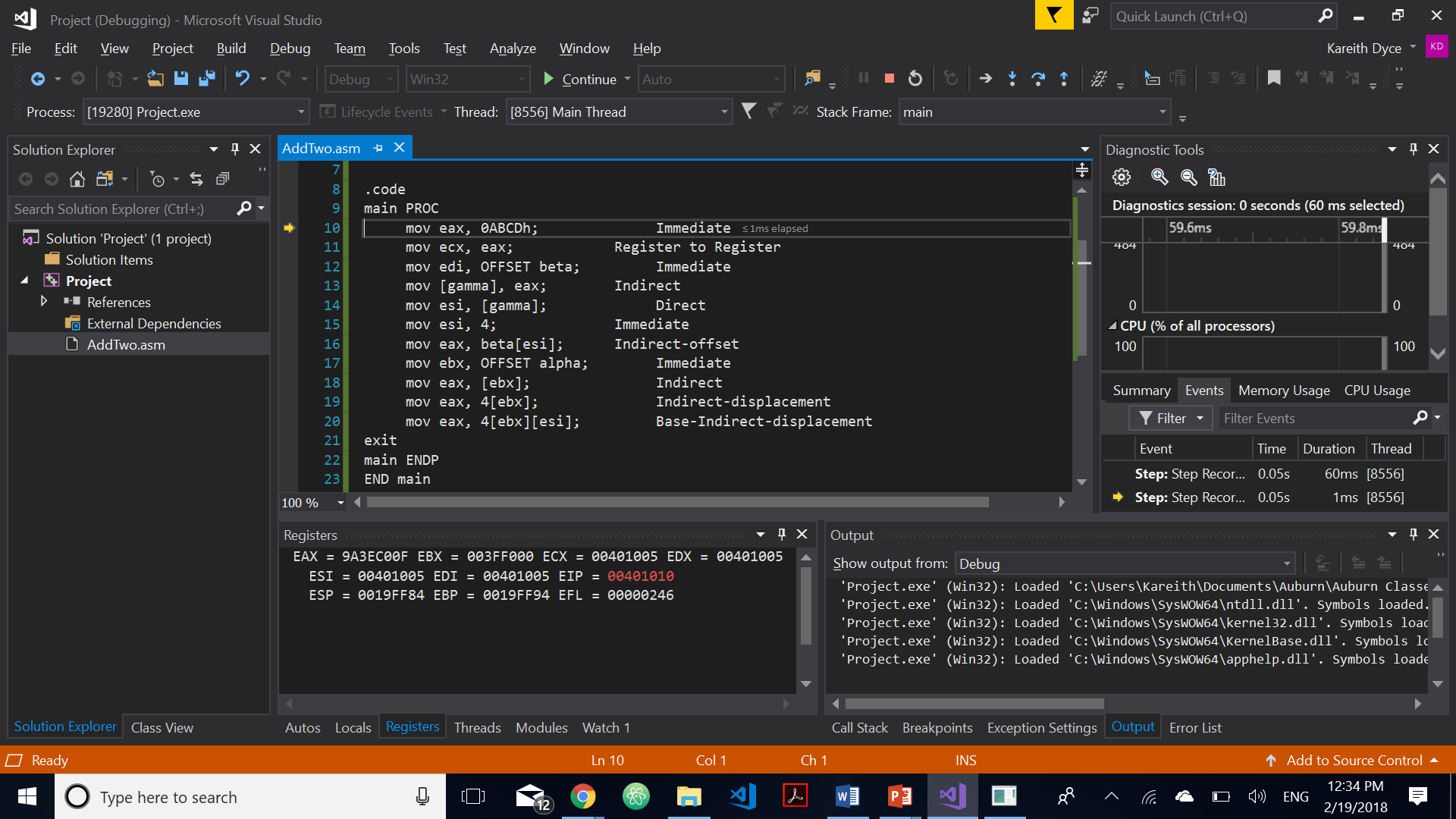
mov eax, 4[ebx]; Indirect-displacement

mov eax, 4[ebx][esi]; Base-Indirect-displacement

exit

main ENDP

END main



Alpha – 00404000

Beta – 00404008

Gamma – 00404010

1. [Indirect addressing] Write a program that subtracts the corresponding elements of Array1 and

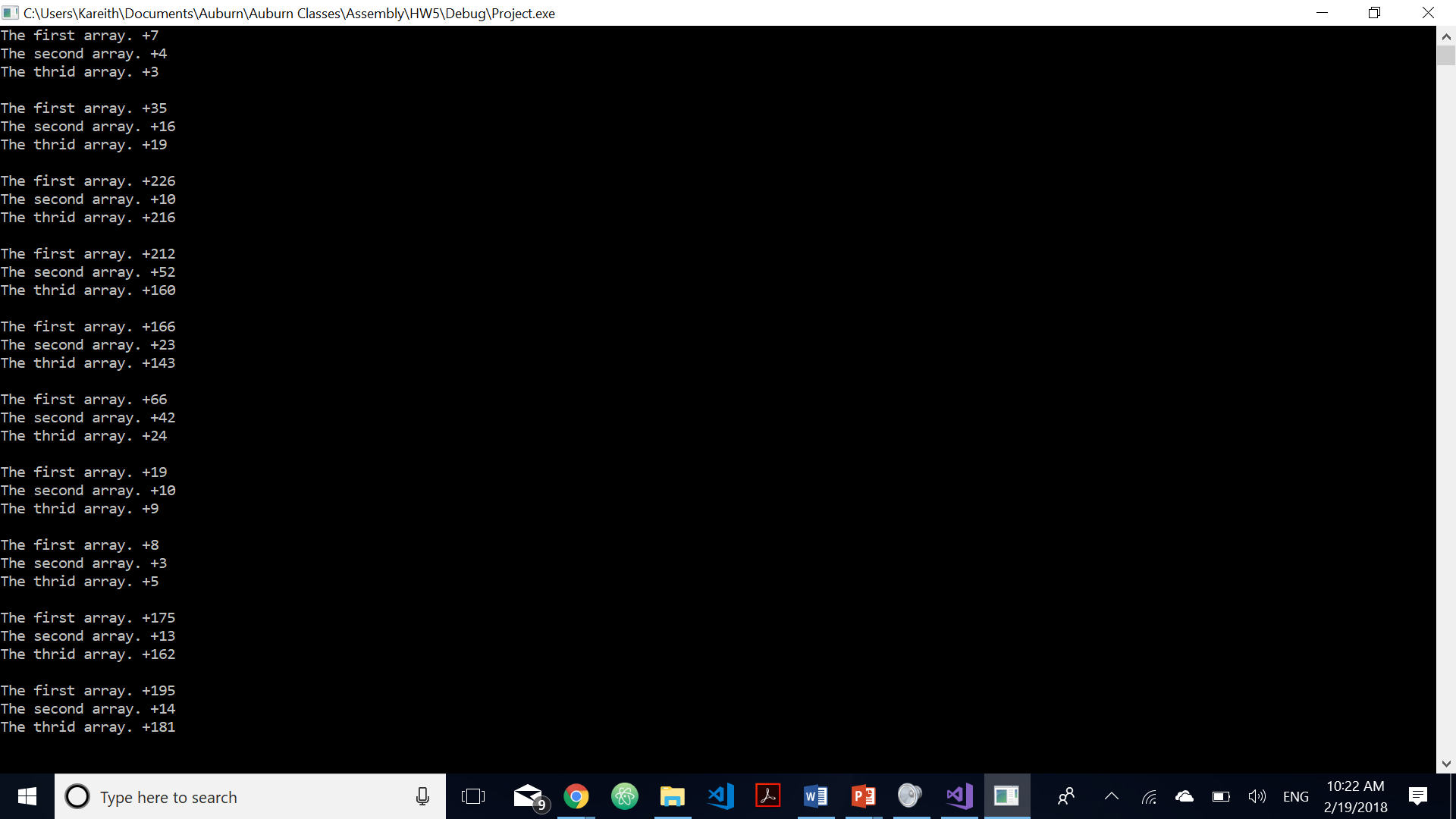
Array2 and stores the results in Array3; e.g. for the 8th element, Array3 [7] 🡨 Array1 [7] - Array2 [7]. Include commands to display the elements of all the arrays. Submit screenshot of the displays of the elements of all the arrays. You can use WriteInt or WriteHex to display the elements of the arrays. Fill in Array1 and Array2 by your own 10 numbers each.

.data

Array1 BYTE 7h, …

Array2 BYTE 4h, …

Array3 BYTE 10 DUP (?)



1. [Loops] Write a program to compute the sum of first *n* odd integers of a series. *Sum = 1 + 3 + 5 +…* Your program must:
   1. Prompt user for integer *n*,
   2. Read the value of *n* from user input
   3. Calculate *Sum*, and;
   4. Print *Sum* on screen.

Please use the “WriteInt” procedure, not “DumpRegs”. Other relevant procedures: “ReadInt” and “WriteString.” The calculation can be done in many ways, and all submissions that evidence proper programming practice are acceptable. In your homework submission, please embed both the code and one screen shot for *n = 15*.

