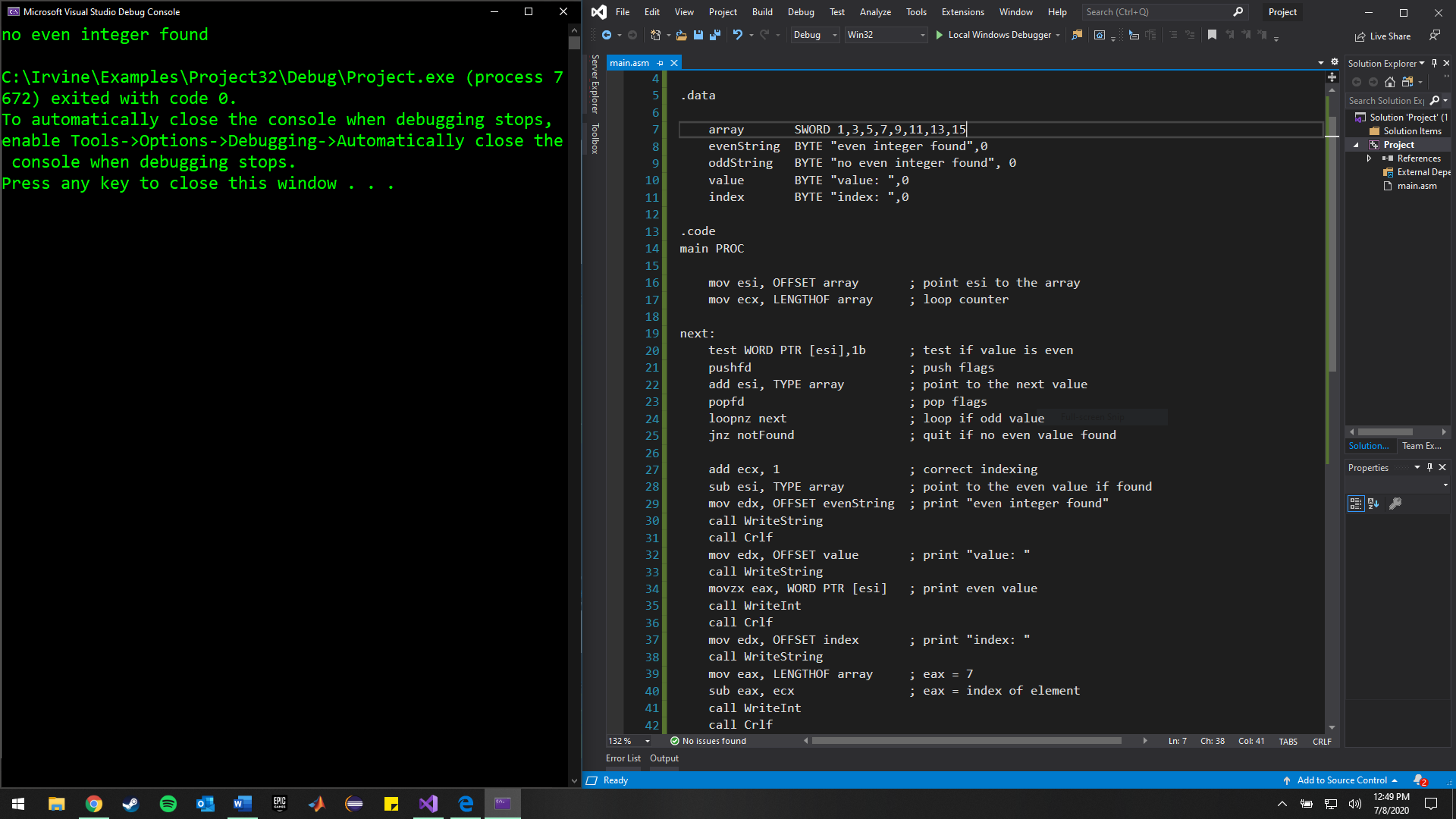
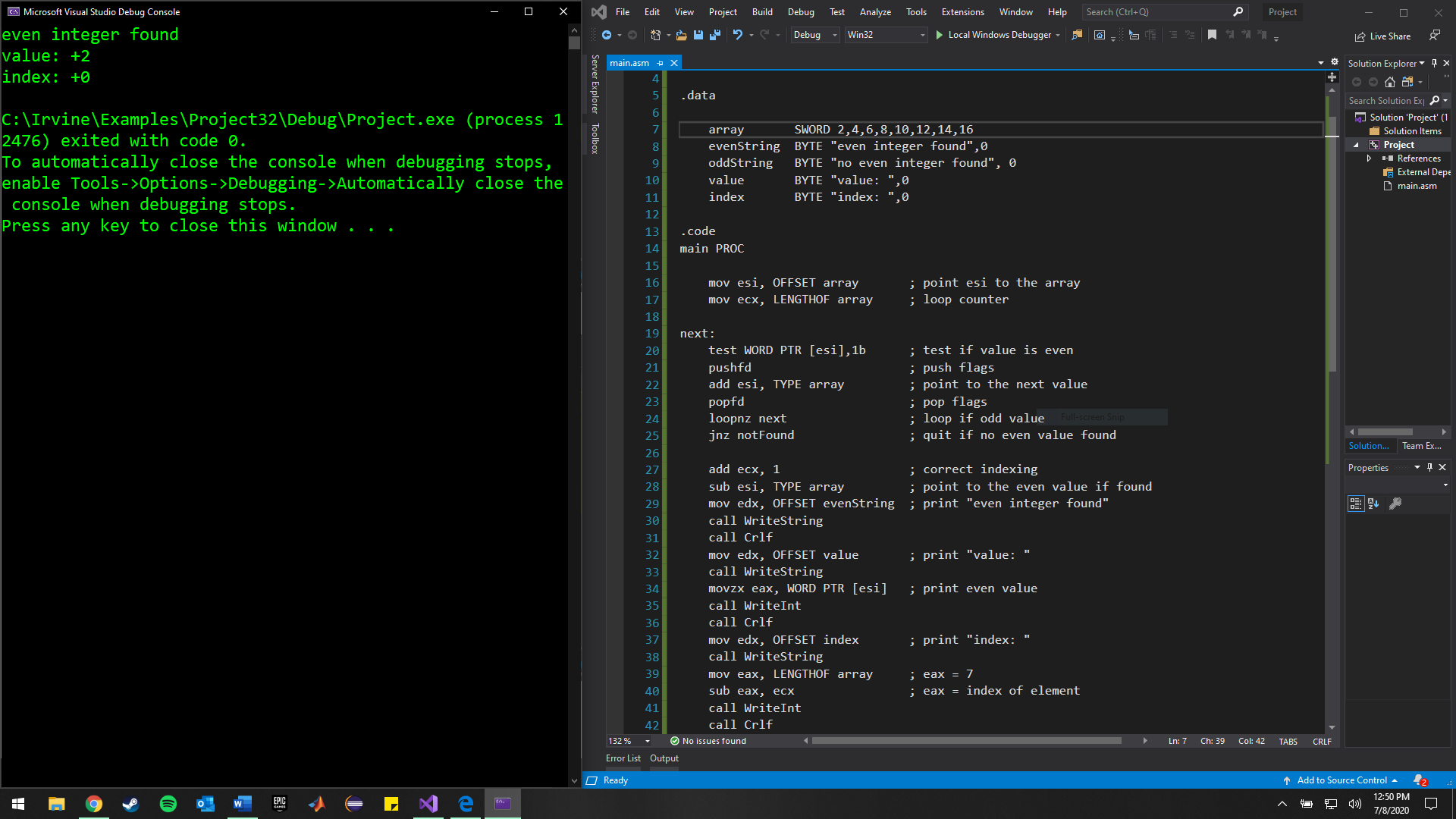
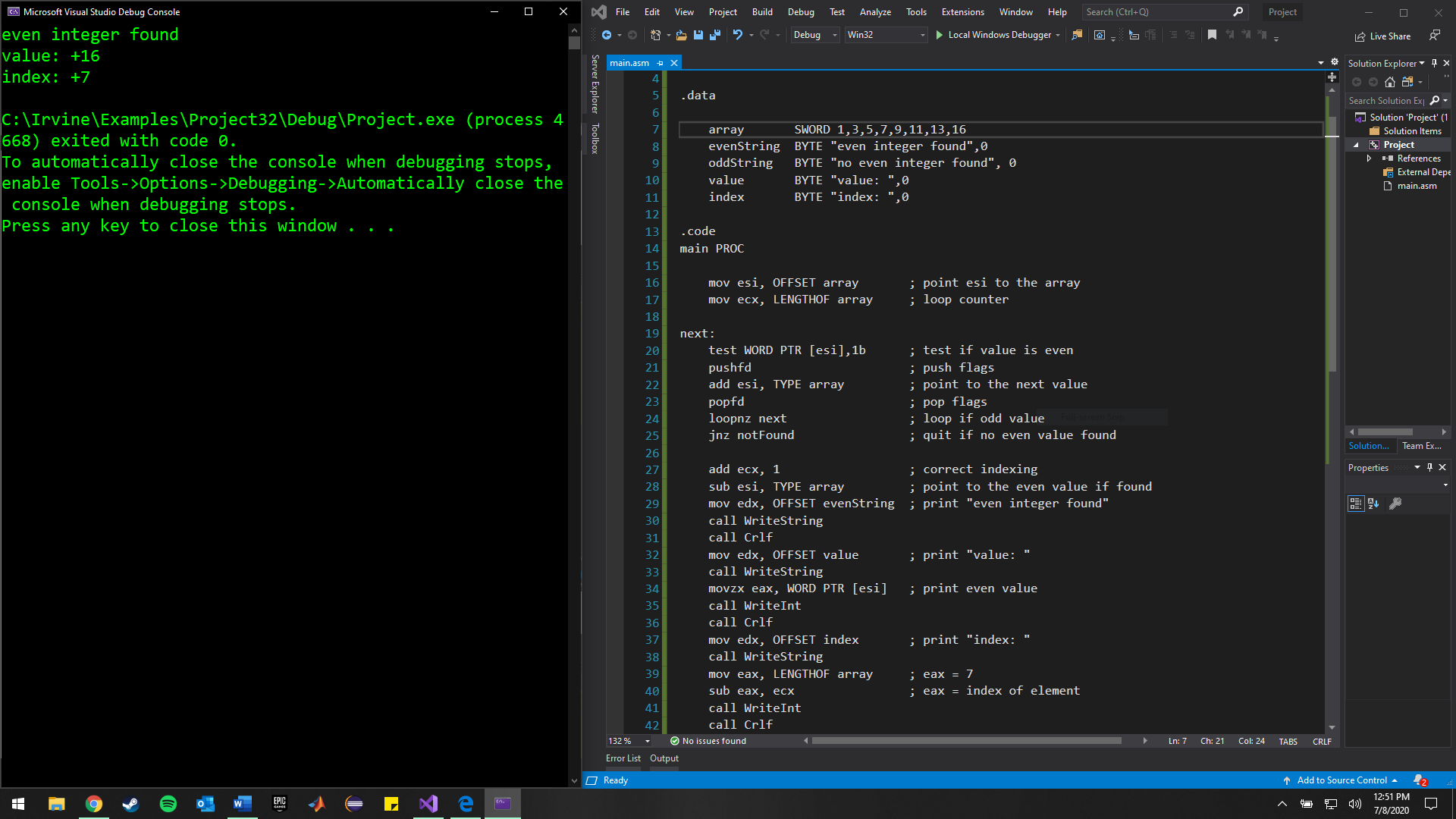
1.



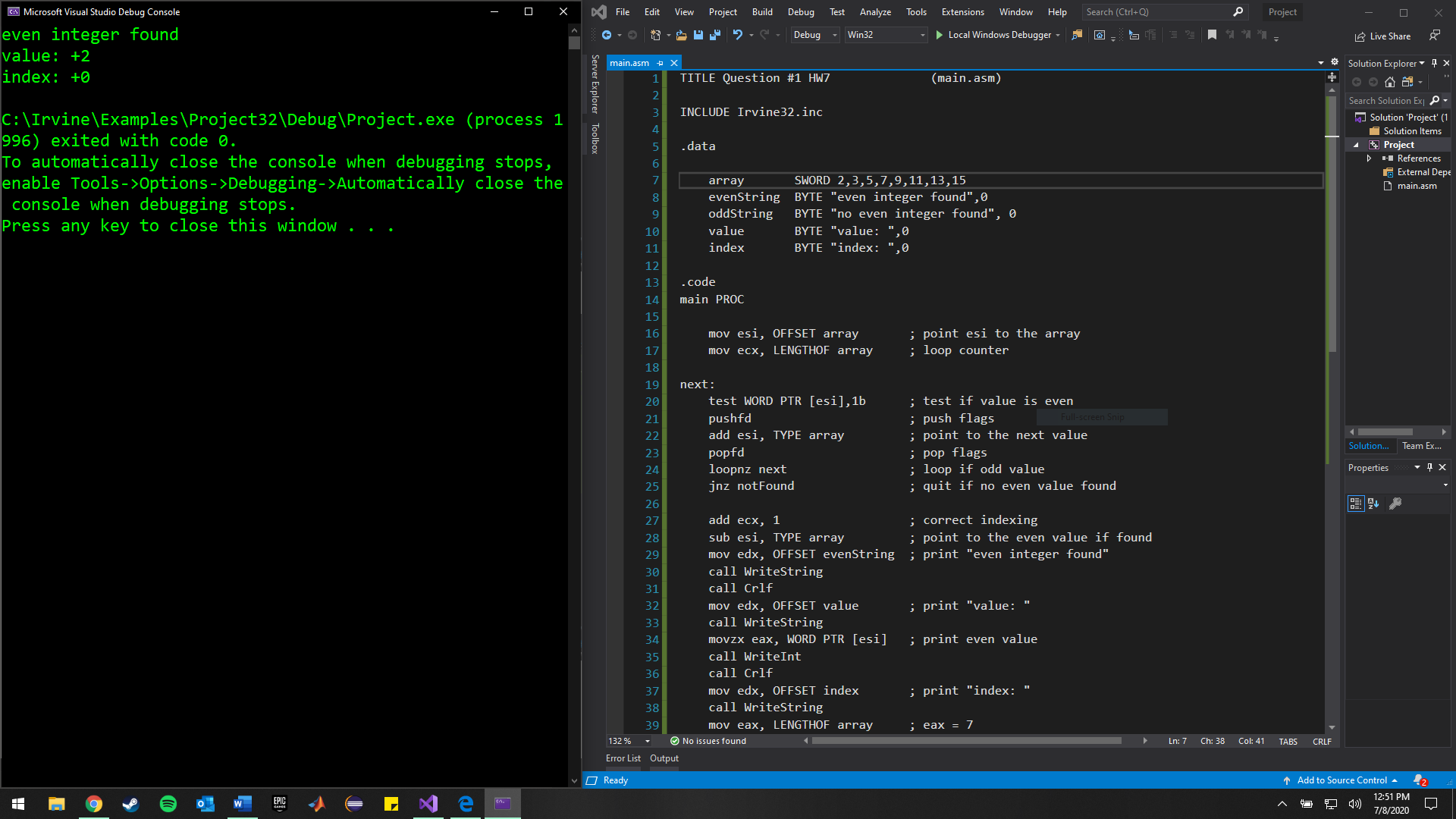
Array has all odd integers.



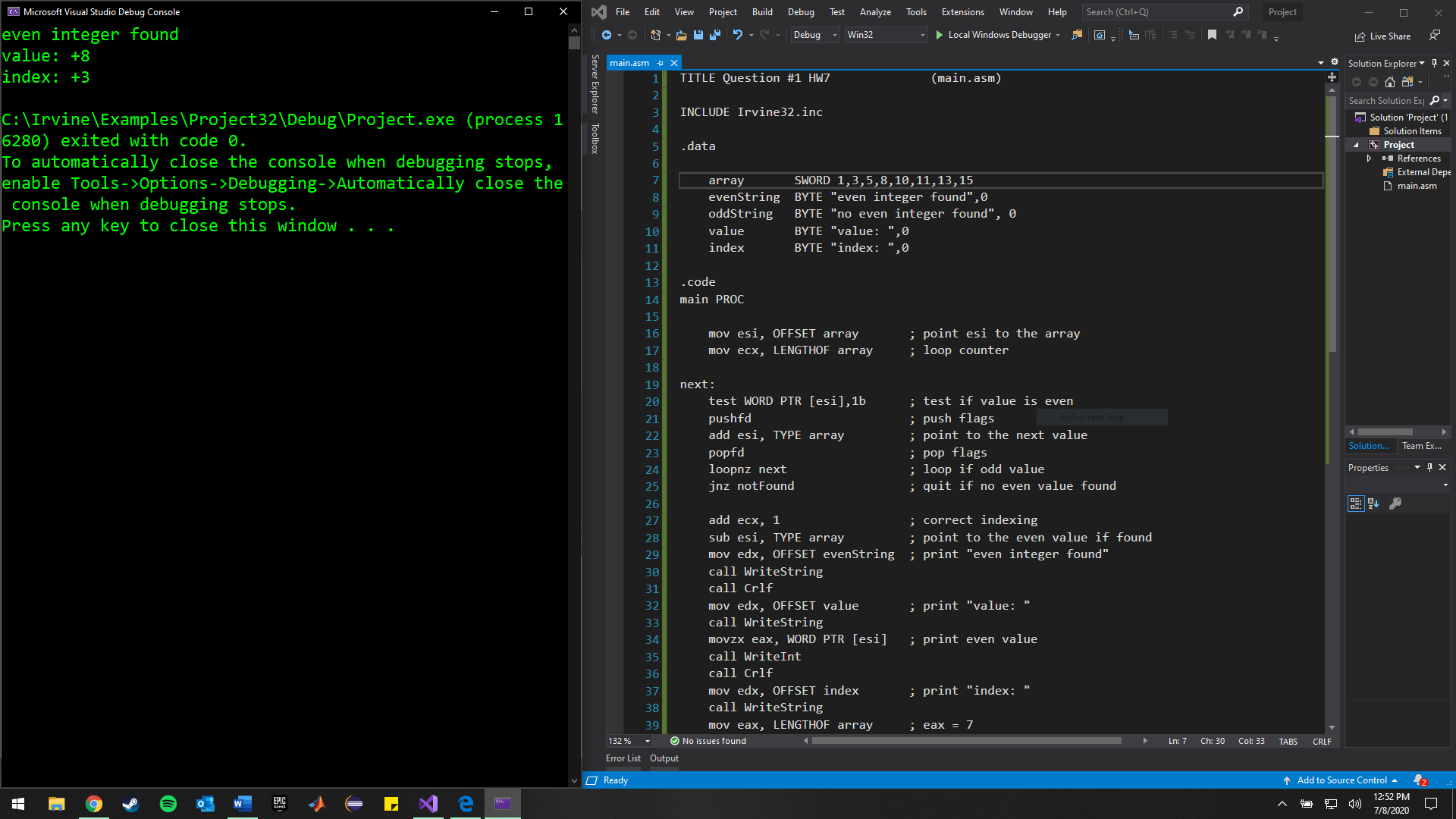
Array has all even integers.



Mixture: even at end of array.

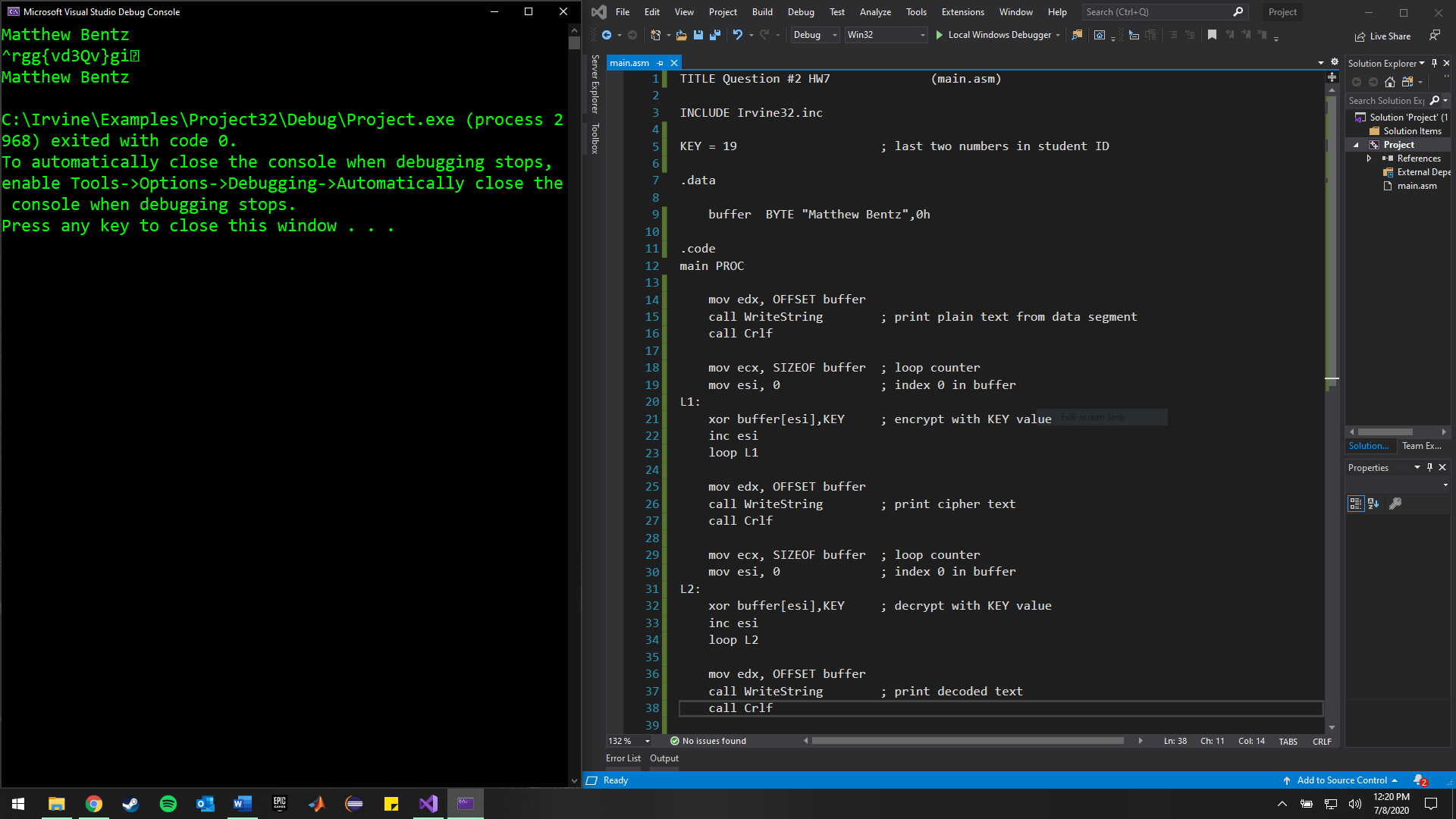


Mixture: even at beginning of array.



Mixture: even in middle of array.

2.



Plain text, cipher text, and decoded text printed in order.

Method strengths: the reversible property of XOR makes it easy to perform since the key value is chosen by the user and that value is used in both encryption and decryption.

Method weaknesses: the encryption can be decoded very easily, especially with short keys. For example, a key of 0 would yield no encryption from the plain text.

3. Testing for different inputs, in order:

1. -70, -70; Bottom interval

2. -41, -41; Both under 40

3. -39, -41; S1 < S2

4. -41, -39; S2 < S1

5. -3, 0; Disagree

6. 0, 0; Agree

7. 0, 3; Disagree

8. 37, 39; S1 < S2

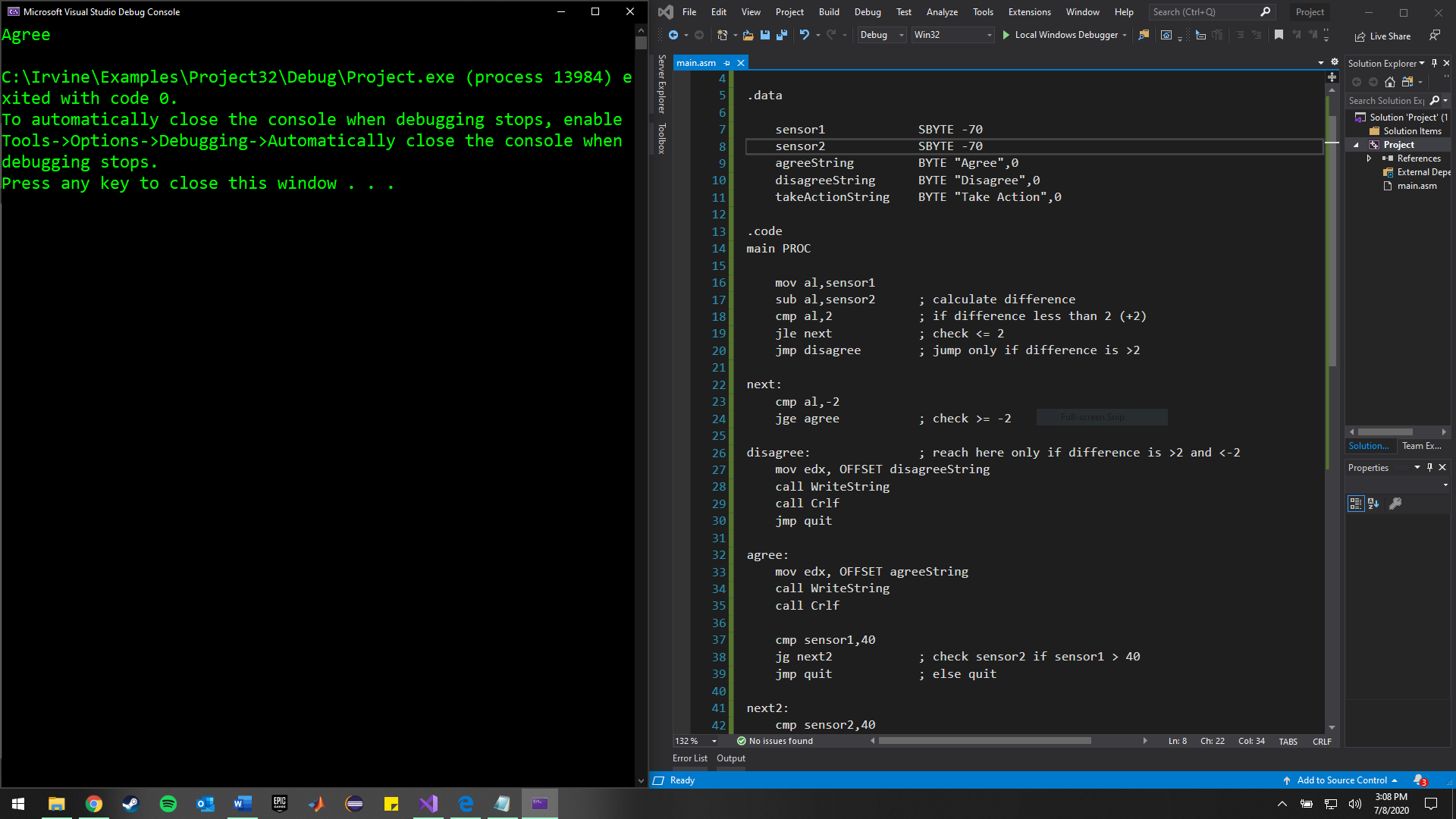
9. 39, 41; S1 < S2 && S2 > 40

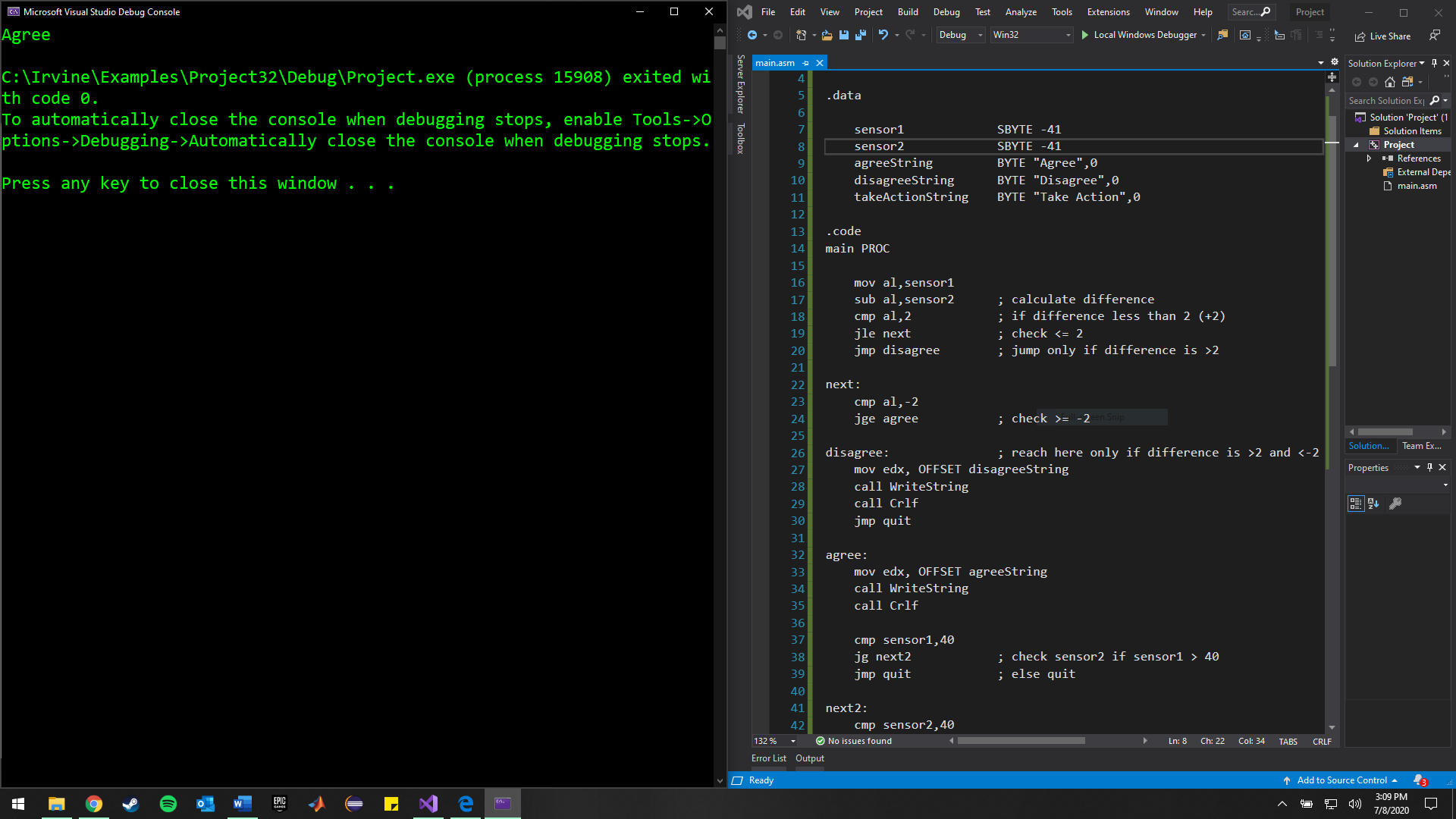
10. 41, 39; S1 > S2 && S1 > 40

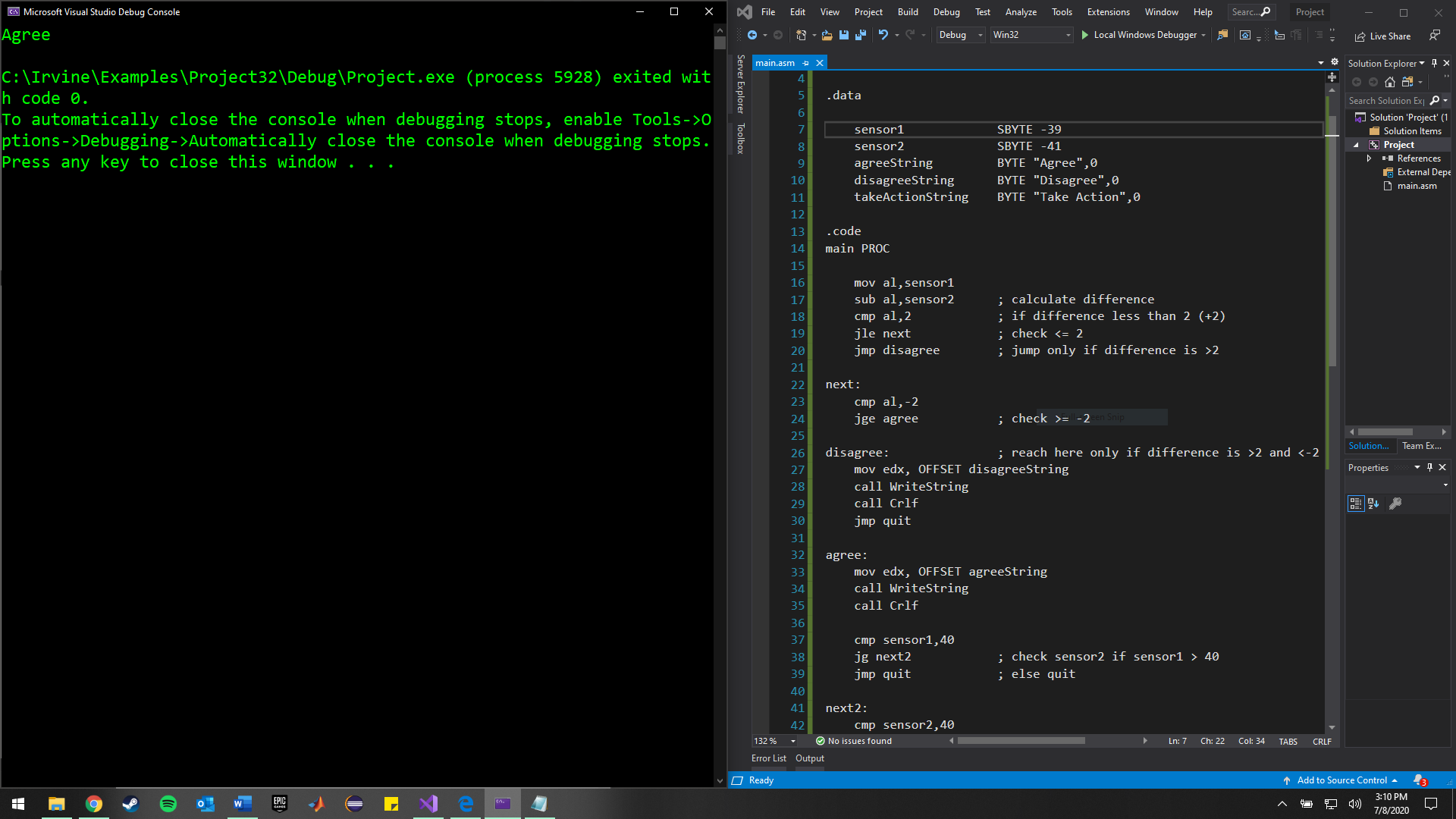
11. 41, 41; S1 = S2 && Both > 40

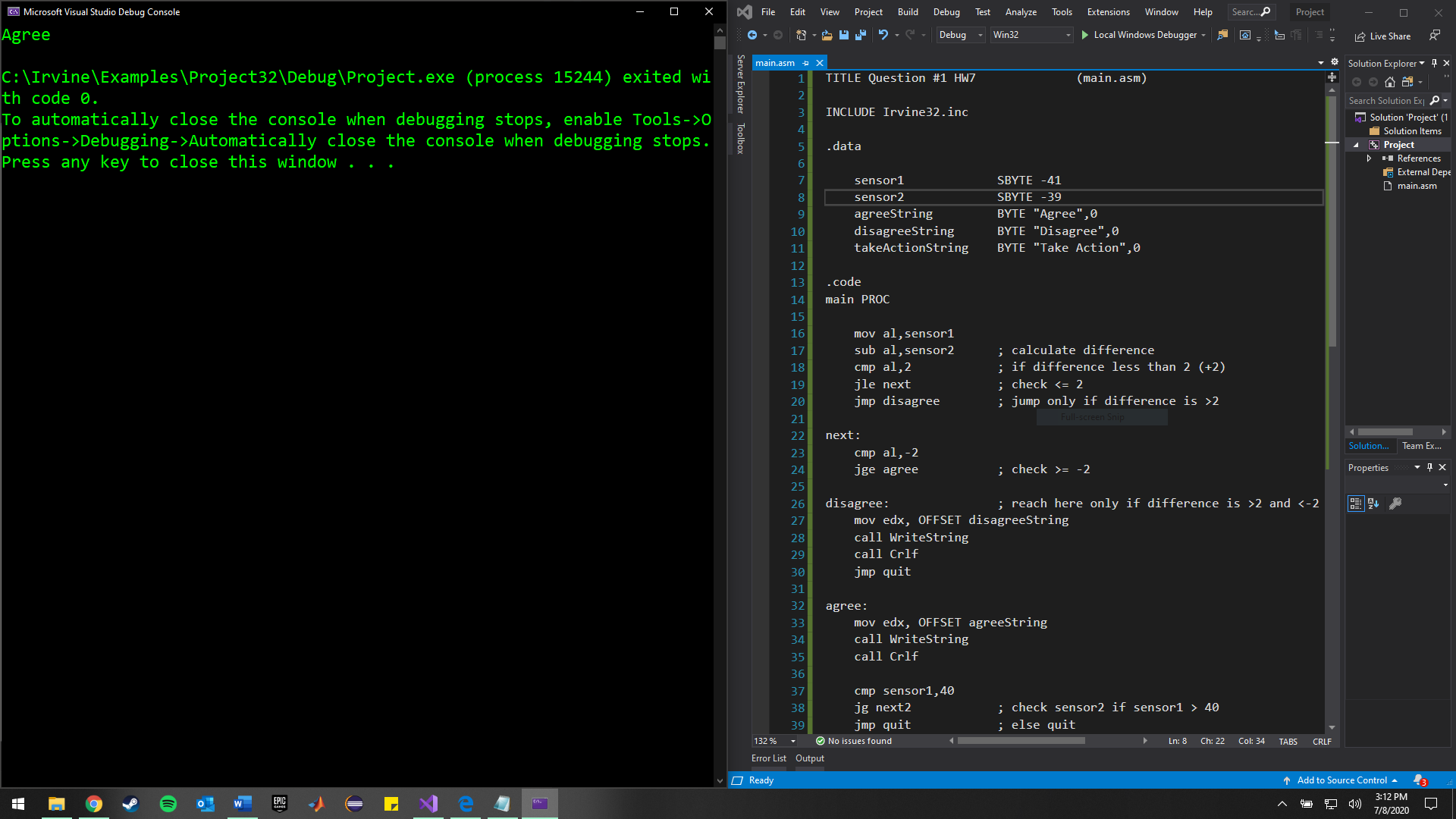
12. 41, 44; Disagree && Both > 40

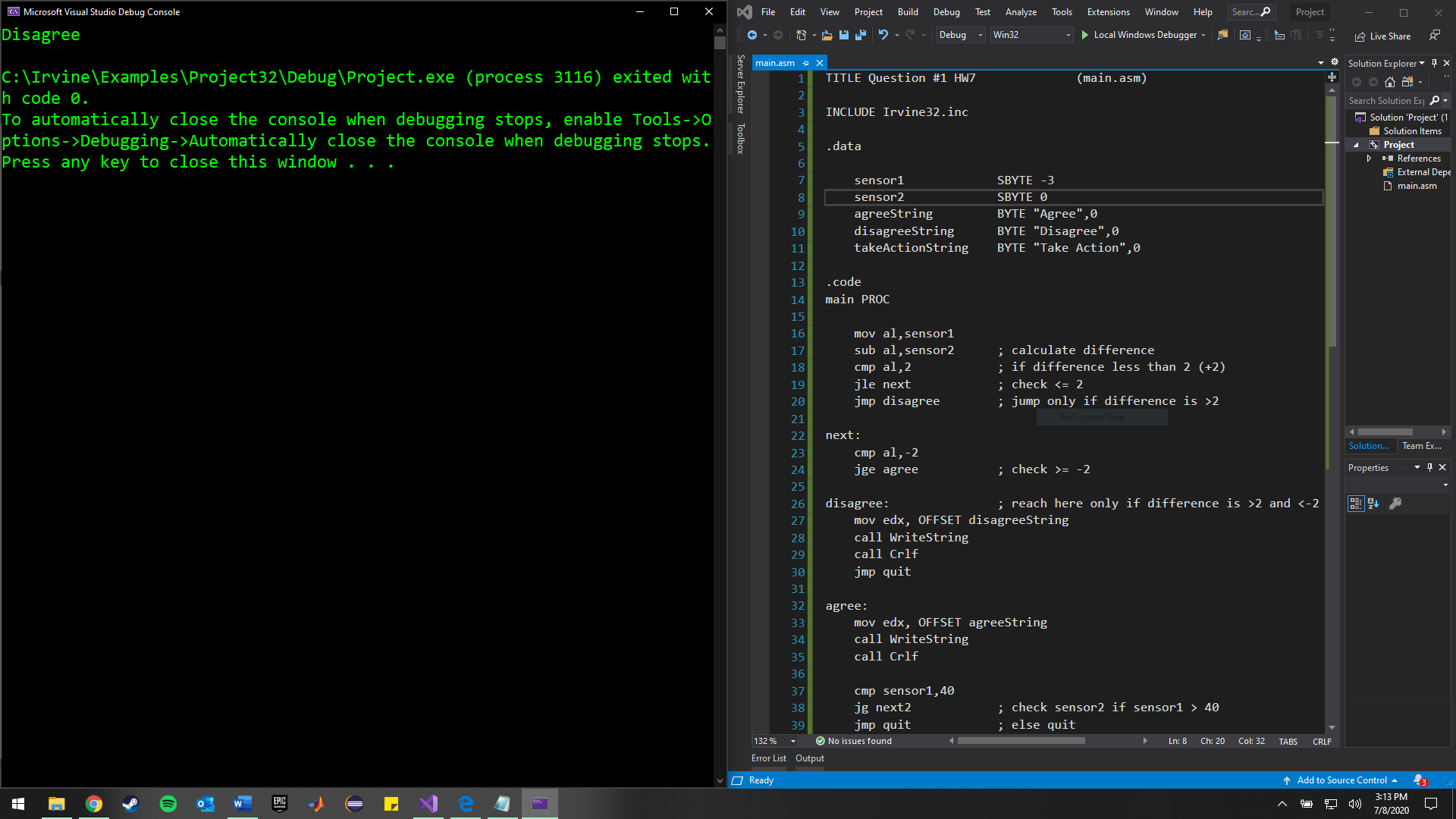
13. 70, 70; Upper interval

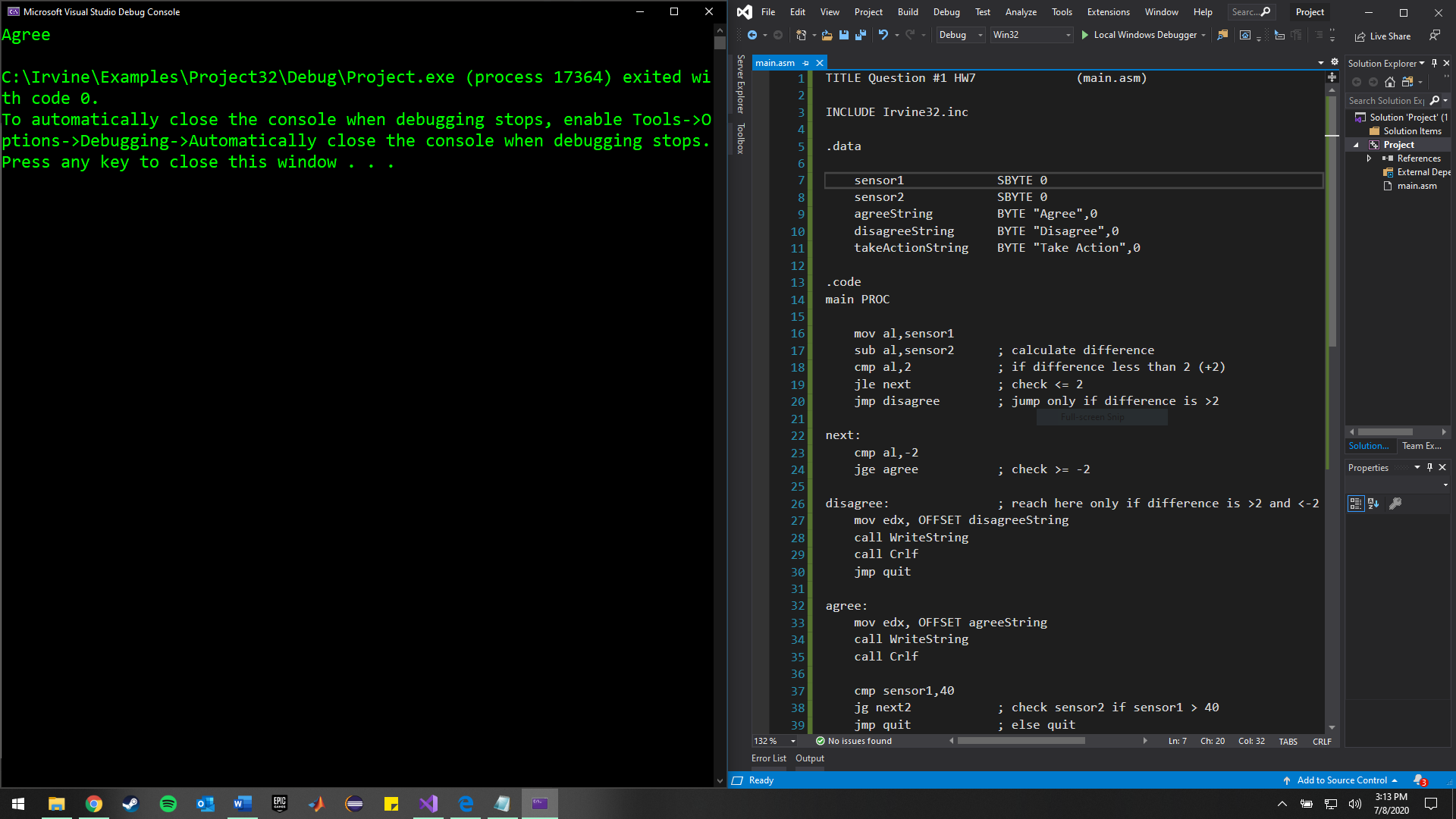


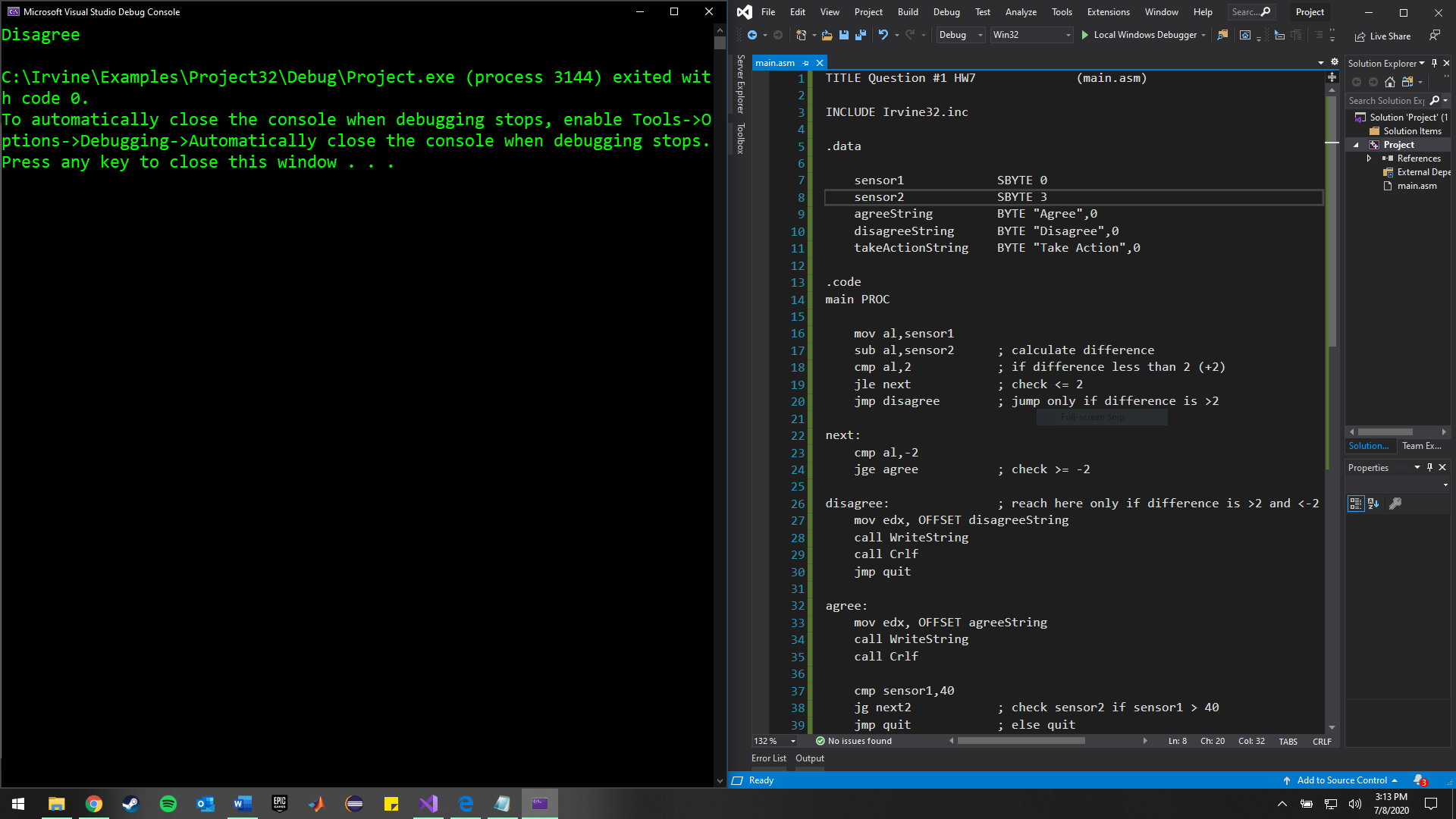


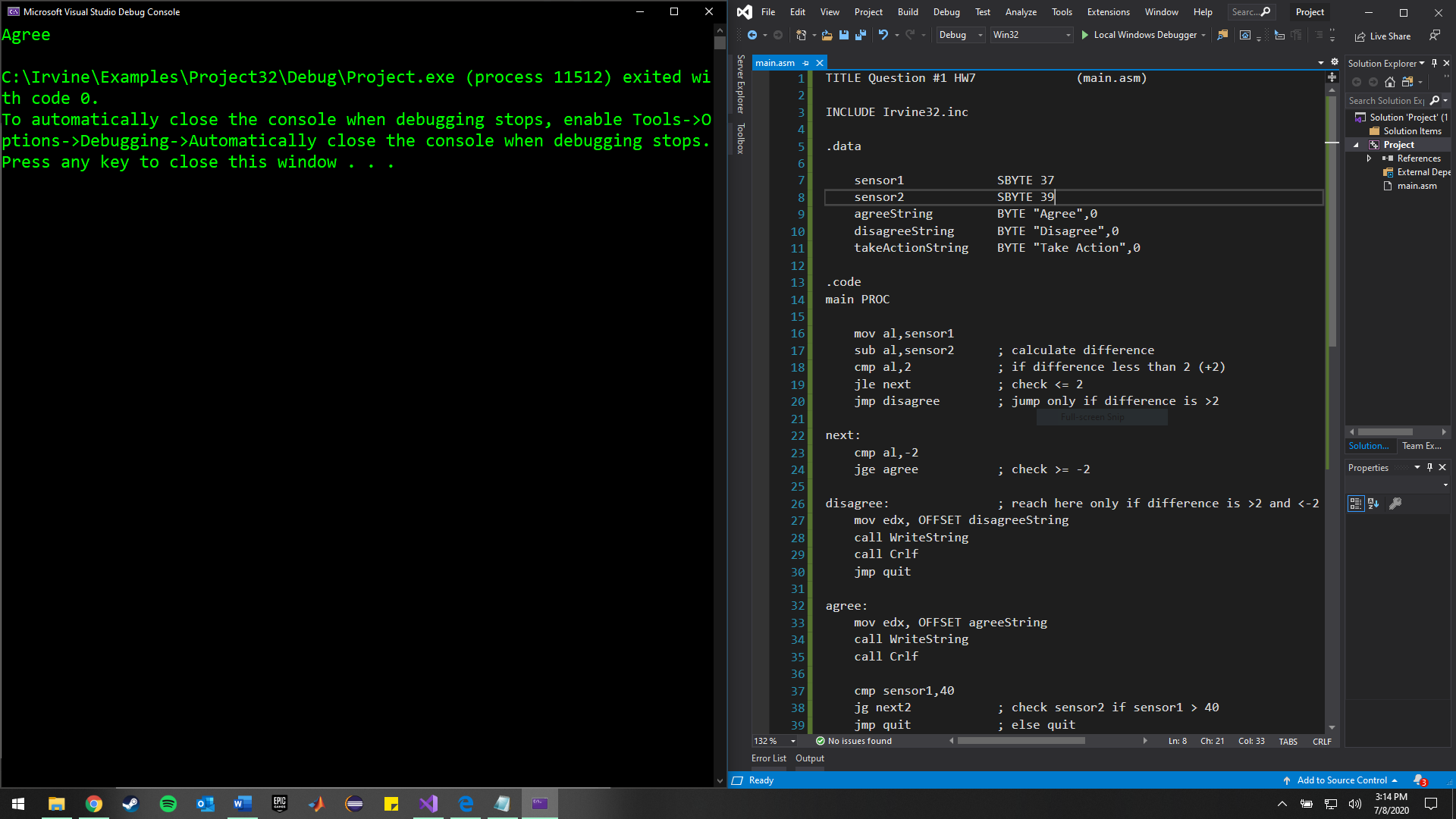


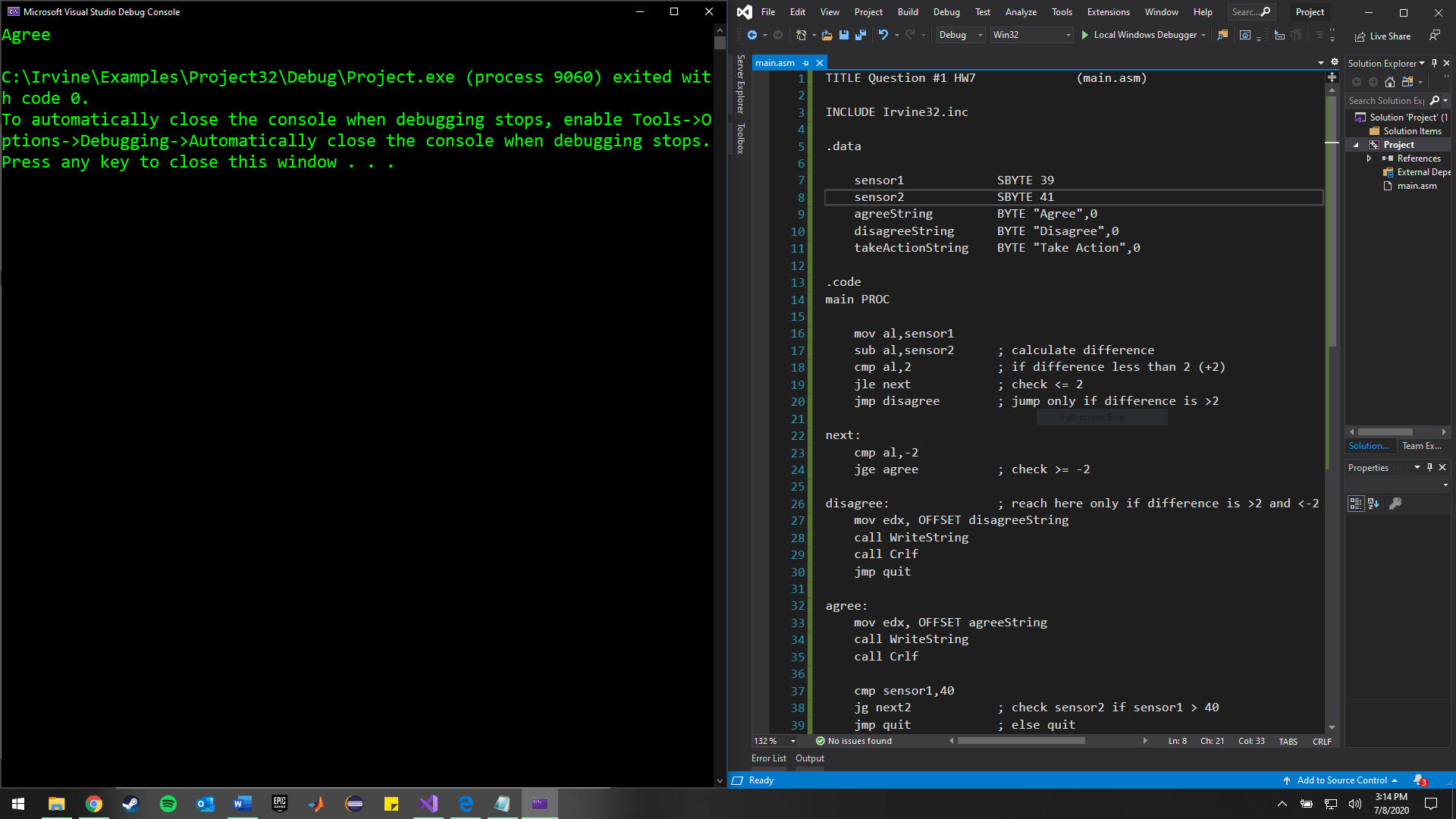


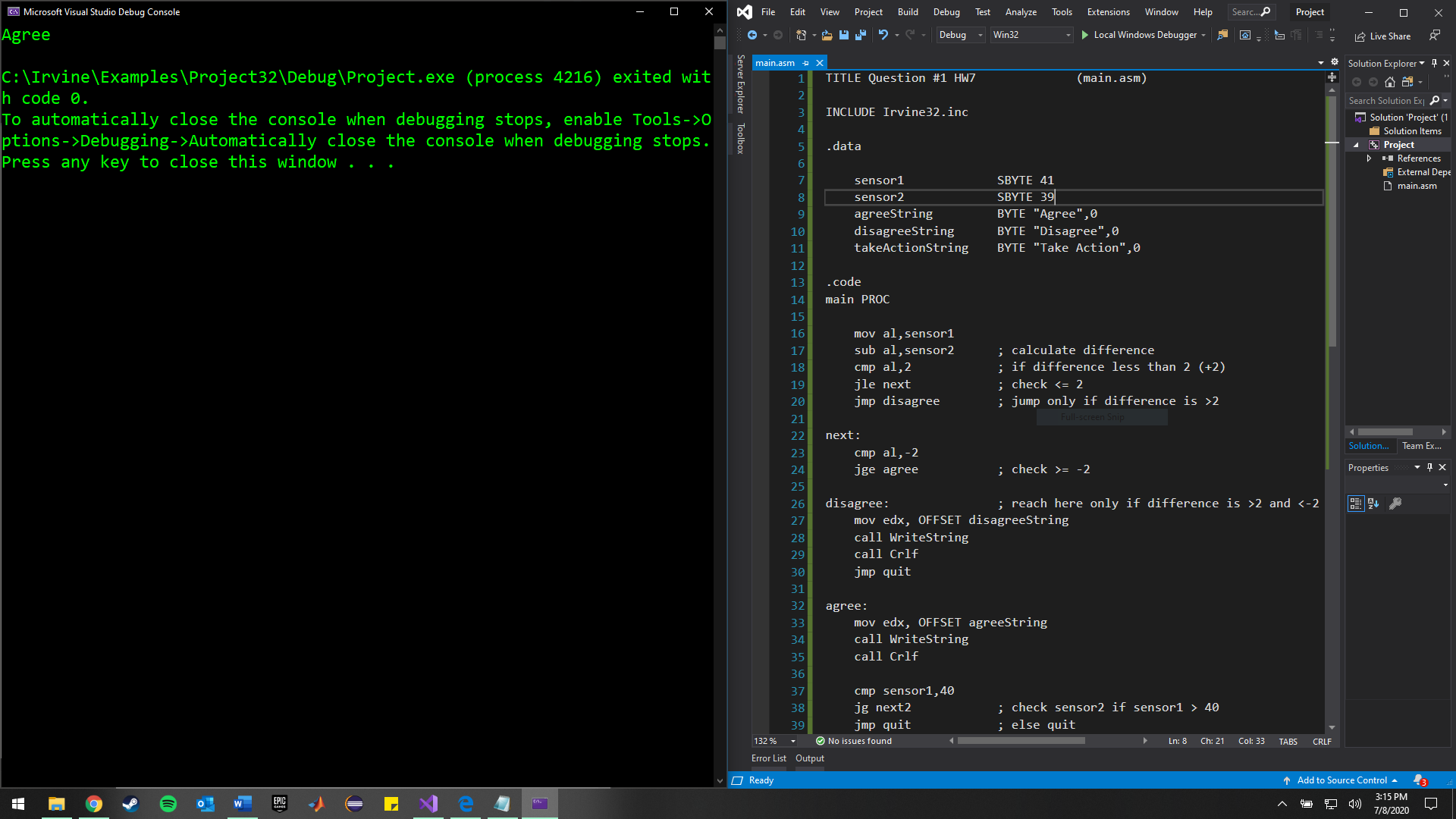


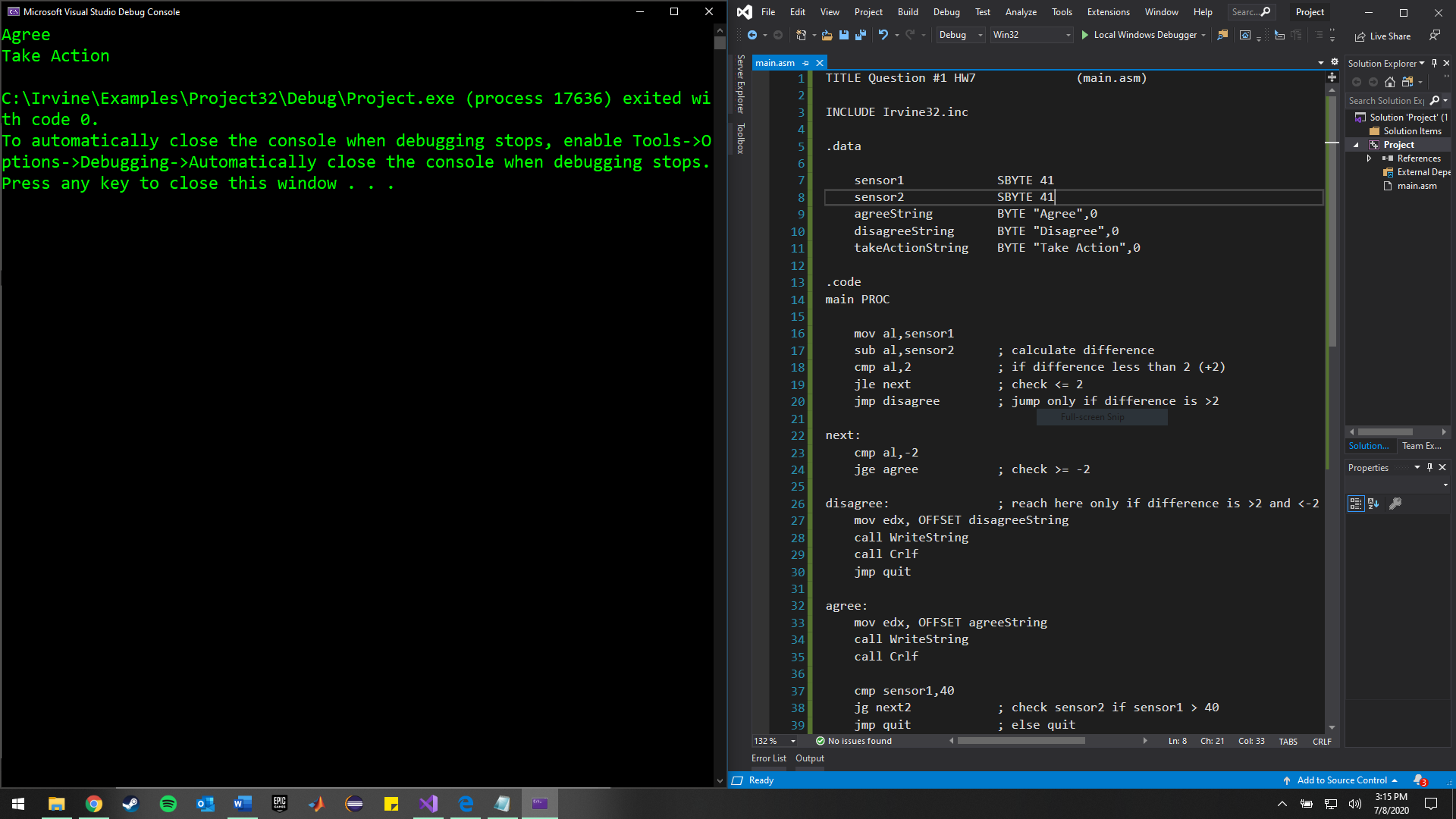


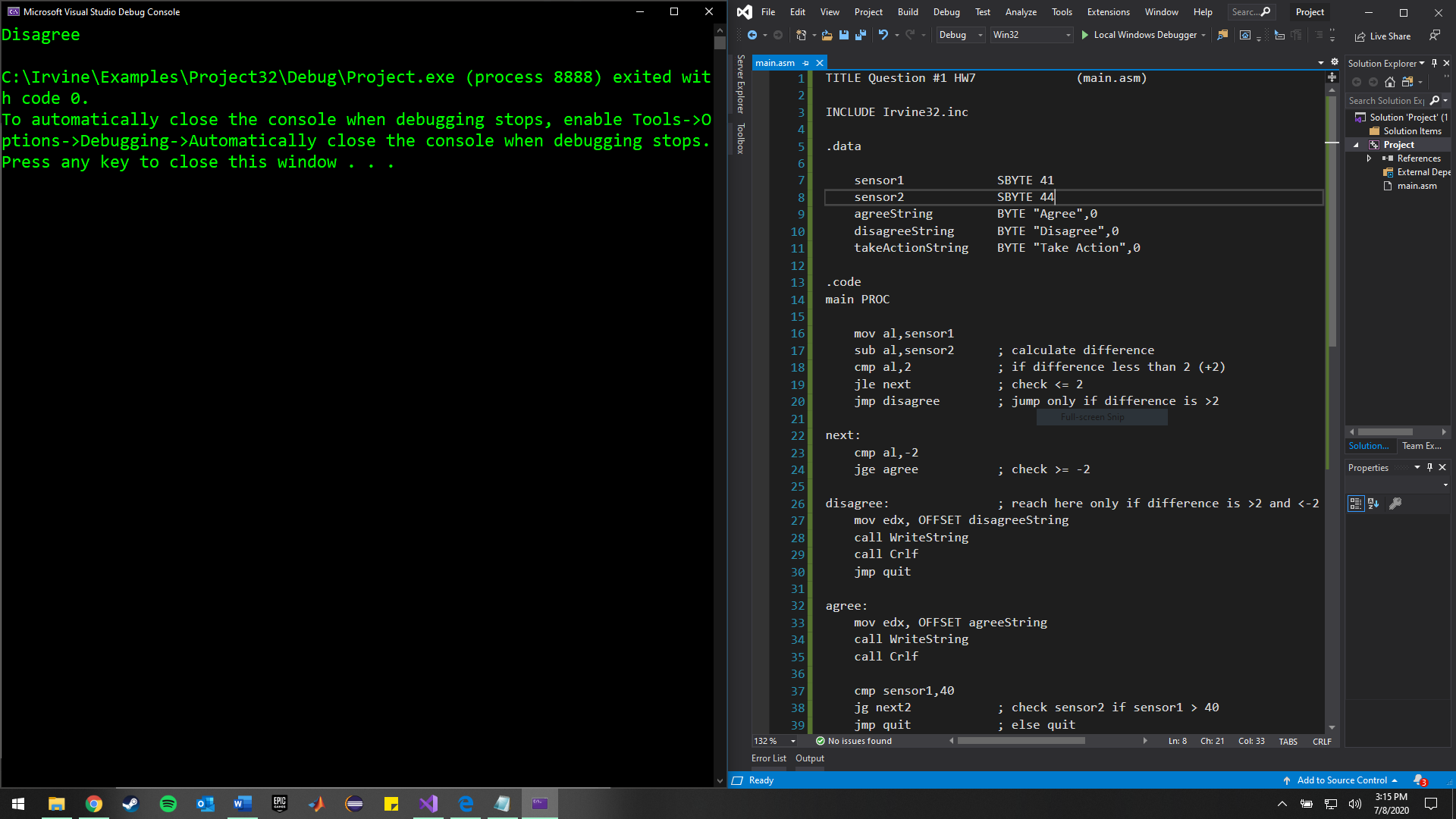


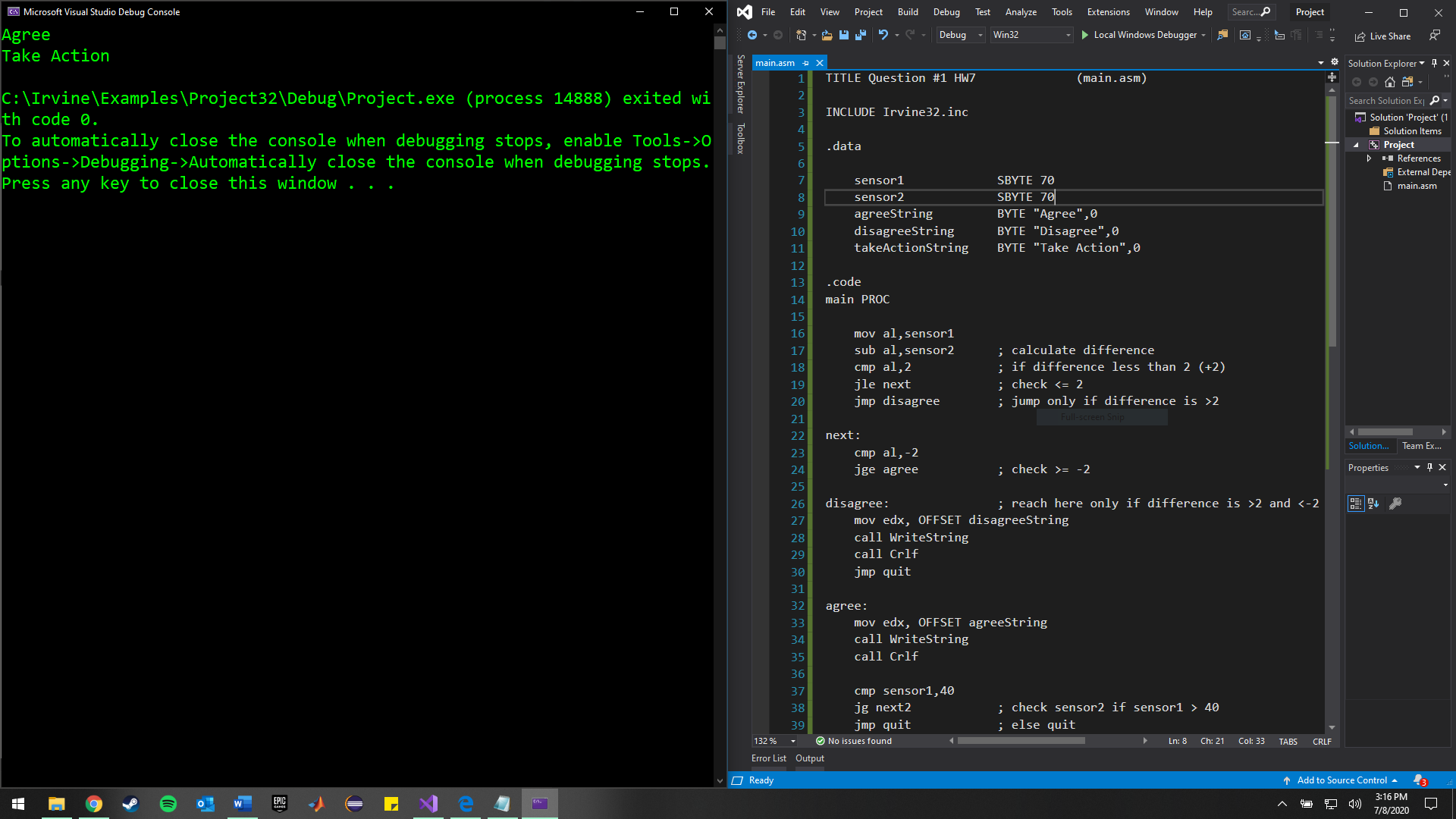












4. Stack is represented from Left -> Right as bottom to top of the stack.

Before call FADD:

Stack = N/A (always the left-most value)

Before Push ecx:

Stack = 4040026 (next instruction offset (EIP) is pushed onto stack), also ESP += 4

Before Push ebx:

Stack = 4040026, 0000000Ch (ecx is pushed onto stack)

Before mov eax, edx:

Stack = 4040026, 0000000Ch, 0000000Bh (ebx is pushed onto stack)

Before Pop ebx:

Stack = 4040026, 0000000Ch, 0000000Bh (mov eax, edx does not affect stack)

Before Pop ecx:

Stack = 4040026, 0000000Ch (ebx = top of stack value)

Before ret:

Stack = 4040026 (ecx = top of stack value)

After ret:

Stack = N/A (EIP value is popped from stack to continue instruction)