CSC 3210

Computer Organization and Programming

Lab 9

Answer Sheet

Student Name: Hakan Gunerli

Section: Lab Section: 014 , CRN - 19377

Lab 9(a)

Debug through each line of instructions.

Take screenshot that includes code and register window.

Record the register content.

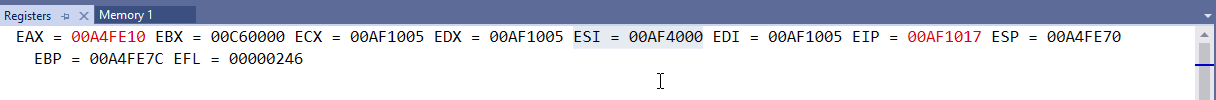
and explain the register contents.

Line number: 13

Instruction: mov esi, OFFSET myBytes

Register values: ESI = 00AF4000

Screenshot:



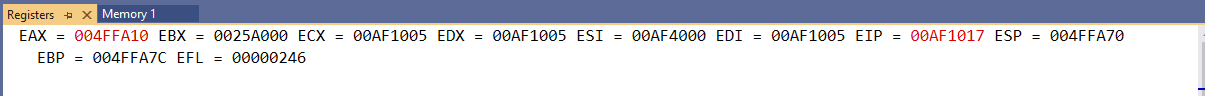
Explanation: The address value of myBytes has been moved to ESI register.

Line number: 14

Instruction: mov al, [esi]

Register values: EAX \_ \_ \_ \_ \_ \_ 10h, ESI = 00AF4000

Screenshot:

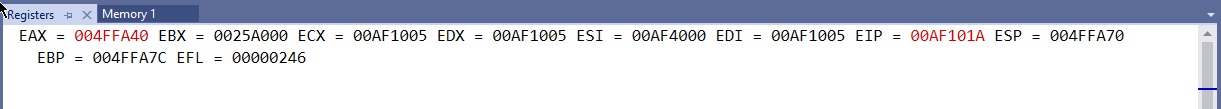


Explanation: first element in myBytes has been moved to AL.

Line number: 15

Instruction: mov al, [esi+3]

Register values: ESI = 00AF4000, EAX = …… 40h

Screenshot: 

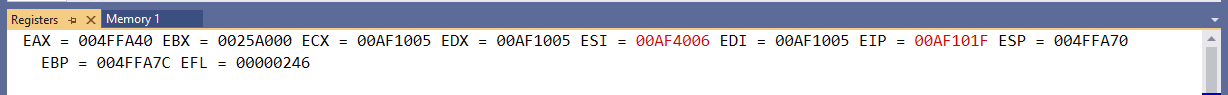
Explanation: last element in myBytes has been moved to AL.

Line number: 16

Instruction: mov esi, OFFSET myWords + 2

Register values: EAX = ……40h, ESI = 00AF4000

Screenshot:

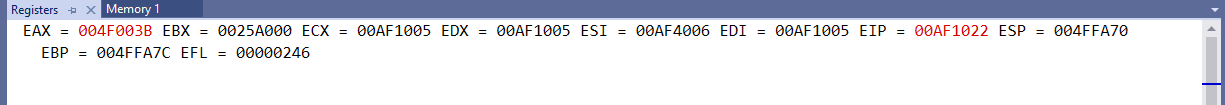


Explanation: The address of the second element in myWords has been moved to AL.

Line number: 17

Instruction: mov ax, [esi]

Register values: EAX 003Bh, ESI 00f4006h

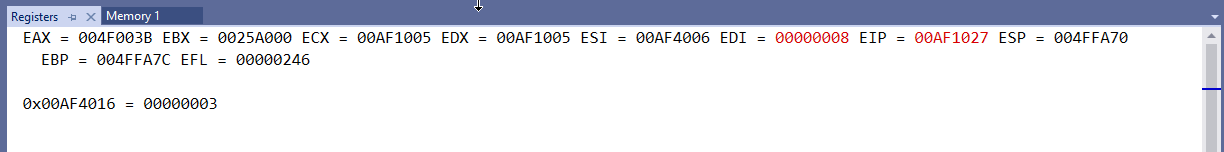
Screenshot: 

Explanation: The second value in myWords has been moved to EAX.

Line number: 18

Instruction: mov edi, 8

Register values: EDI = 00000008h, eax 003Bh

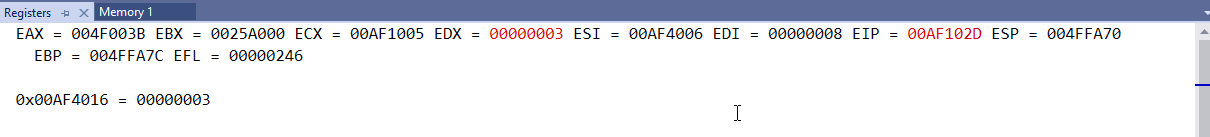
Screenshot: 

Explanation: The value of 8 is moved to edi

Line number: 19

Instruction: mov edx, [myDoubles+edi]

Register values: edx 3h, edi 8h, esi 00af4006h, eax 003Bh

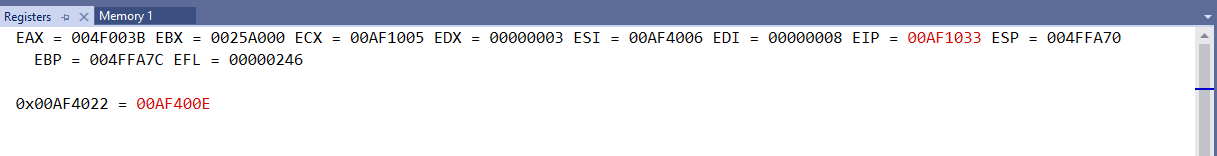
Screenshot: 

Explanation: The third element in mydoubles array has been moved to edx.

Line number: 20

Instruction: mov edx, myDoubles[edi]

Register values: EDI=8h, EDX=3h, eax= 0003Bh, edi 00AF4006

Screenshot: 

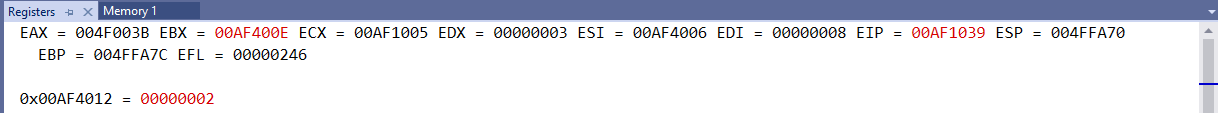
Explanation: no change in the value of registers, since the same element is moved to edx.

Line number: 21

Instruction: mov ebx, myPointer

Register values: EAX=003B, EBX 00AF400E, EDX 3h, ESI=00Af4006, EDI 00000008

Screenshot:

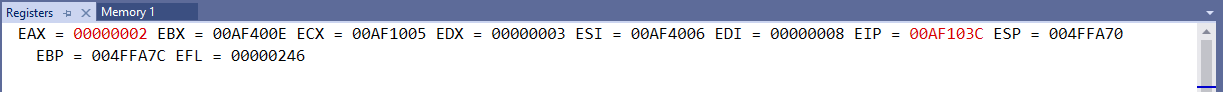


Explanation: the memory address of mydoubles is moved to ebx.

Line number: 22

Instruction: mov eax, [ebx+4]

Register values: EAX=00000002, EBX 00AF400E, EDX 3h, ESI00AF103C, EDI 00000008

Screenshot: 

Explanation: The second element in mydoubles is moved to eax, and the address in ebx is accessed after 4 is added.

Lab 9(b)

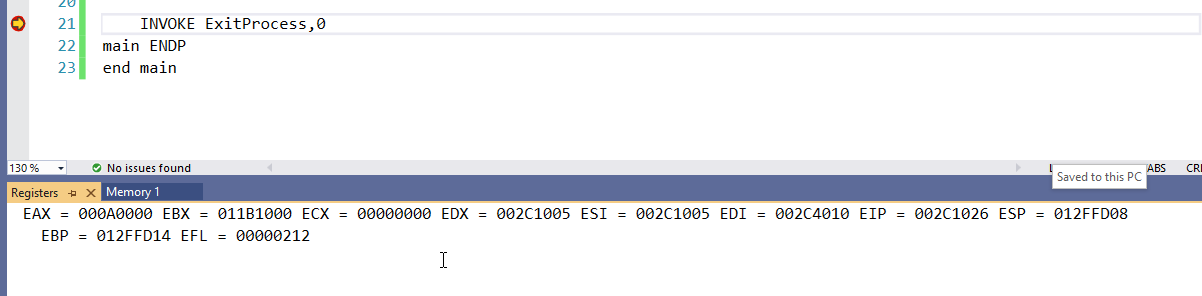
Debug until you reach “INVOKE ExitProcess, 0”.

Take a screenshot of the code and register window at the end

Record the content of the EAX register

**Then explain the register content.**

EAX Register value: EAX 000A0000h

Screenshot: 

Explanation:

The values in intarray are all added,

00010000h

00020000h

00030000h

00040000h

In total, 000A0000h

Lab 9(c)

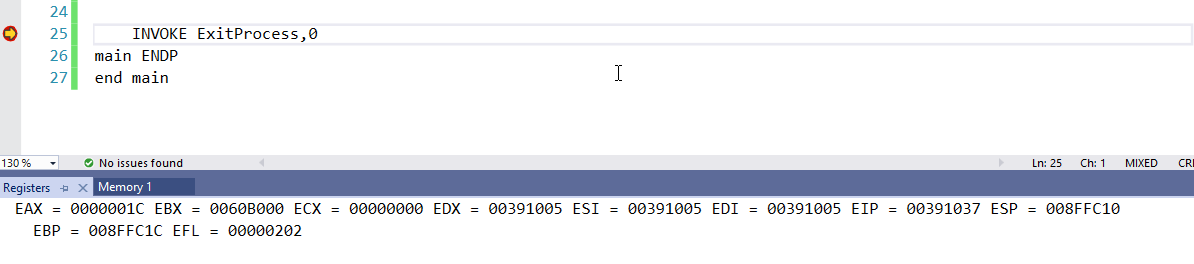
Debug until you reach “INVOKE ExitProcess, 0”.

Take a screenshot of the code and register window at the end

Record the content of the EAX register

**Then explain the register content.**

EAX Register value: EAX= 0000001C hex

Screenshot: 

Explanation:

The loop inside goes 5 times of every iteration of 10 loops. For each iteration (5 iterations), 5 is added to three. In every iteration, EAX is reset but at the end, we end up with 28d which is 1C.

Lab 9(d)

Debug through each line of instructions.

Take screenshot that includes code and register window.

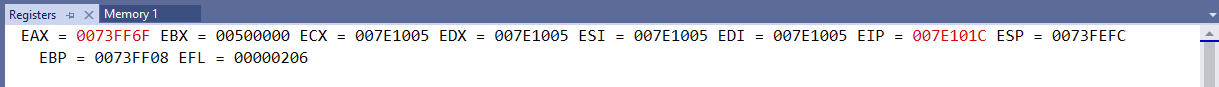
Record the register content.

and explain the register contents.

Line number: 9

Instruction: mov al, 01101111b

Register values: EAX= 6Fh

Screenshot: 

Explanation:

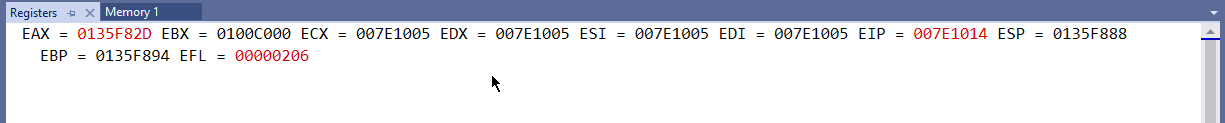
The hex value of 01101111b is moved to AL register as hex.

Line number: 10

Instruction: and al, 00101101b

Register values: EAX= ….2Dh

Screenshot:



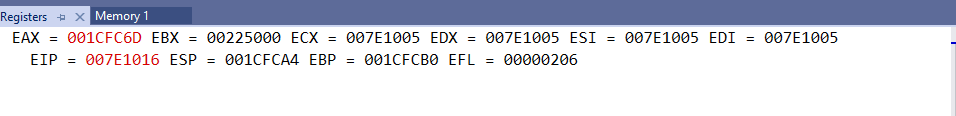
Explanation:

The bits that are the same in both areas are carried, and others are replaced with zeros. [and operator]

Line number: 11

Instruction: mov al, 6Dh

Register values: EAX = ……6Dh

Screenshot: 

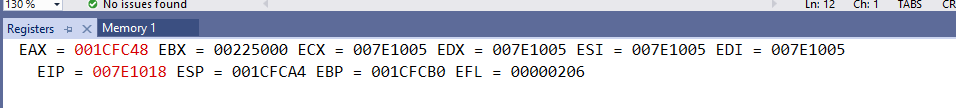
Explanation:

Value of 6D is moved to AL.

Line number: 12

Instruction: and al, 4Ah

Register values: EAX……48h

Screenshot: 

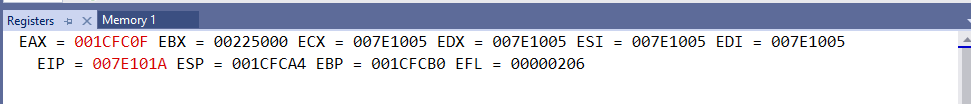
Explanation: AND has been performed between two numbers, the same digits are carried and others are 0.

Line number: 13

Instruction: mov al, 00001111b

Register values: EAX ……0Fh

Screenshot:



Explanation:

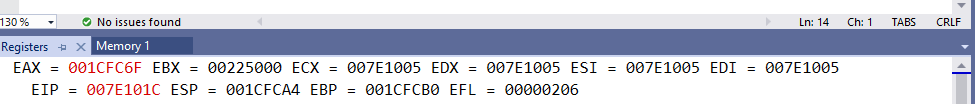
Value for 15 binary is moved to eax.

Line number: 14

Instruction: or al, 61h

Register values: ……6F

Screenshot:



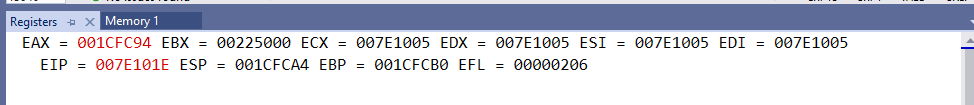
Explanation:

Values with larger numbers of 1 bits are carried over due to being an OR instruction.

Line number: 15

Instruction: mov al, 94h

Register values: ……94h

Screenshot: 

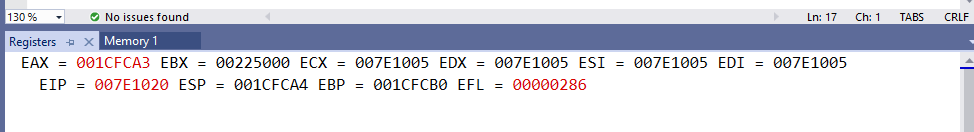
Explanation:

Value of 94h moved to eax.

Line number: 16

Instruction: xor al, 37h

Register values: EAX= A3h

Screenshot: 

Explanation:   
xor operation with 2 numbers, every bit with different value replaced with 1 and same values are replaced with 0 bits.