CSC 3210

Computer Organization and Programming

Lab 7

Answer Sheet

Student Name: Aparna Mandapaka

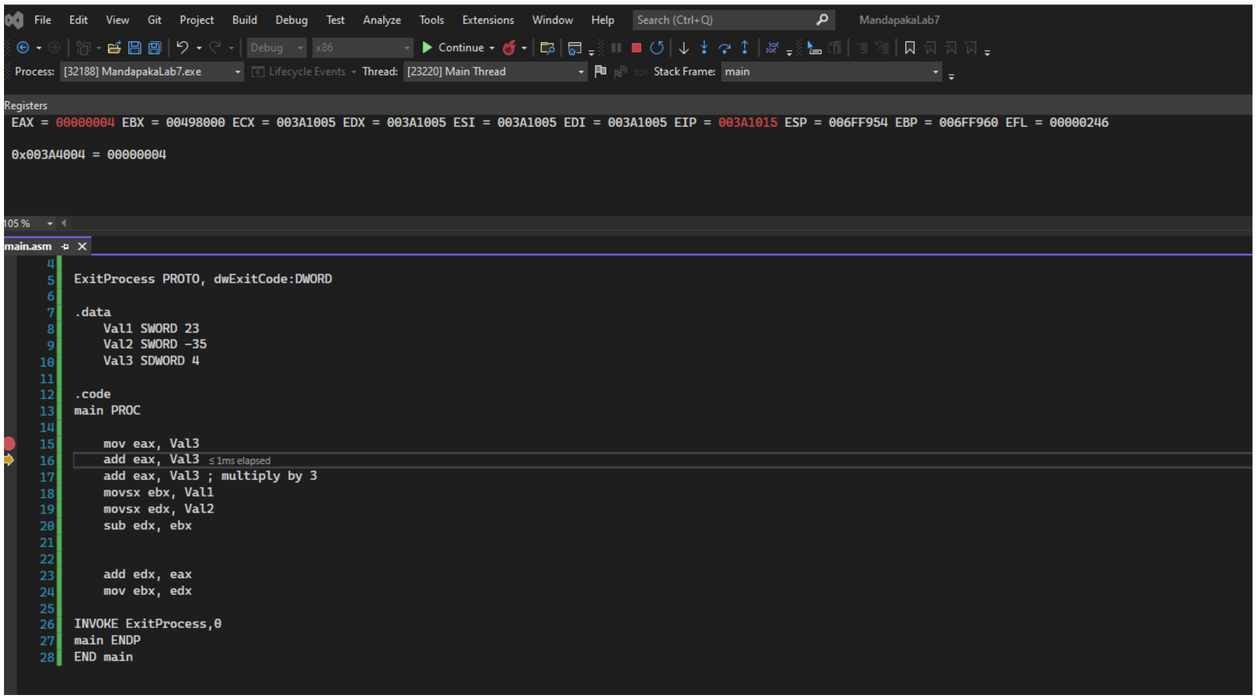
Section:

Debug through each line of code and explain the register content.

Line: 15

Instruction: mov eax, Val3

Register value: EAX = 00000004


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
MandapakaLab7
Debug
x86
Continue
Process: [32188] MandapakaLab7.exe
Lifecycle Events
Thread: [23220] Main Thread
Stack Frame: main
Registers
105 %
main.asm

```
EAX = 00000004 EBX = 00498000 ECX = 003A1005 EDX = 003A1005 ESI = 003A1005 EDI = 003A1005 EIP = 003A1015 ESP = 006FF954 EBP = 006FF960 EFL = 00000246

0x003A4004 = 00000004
```

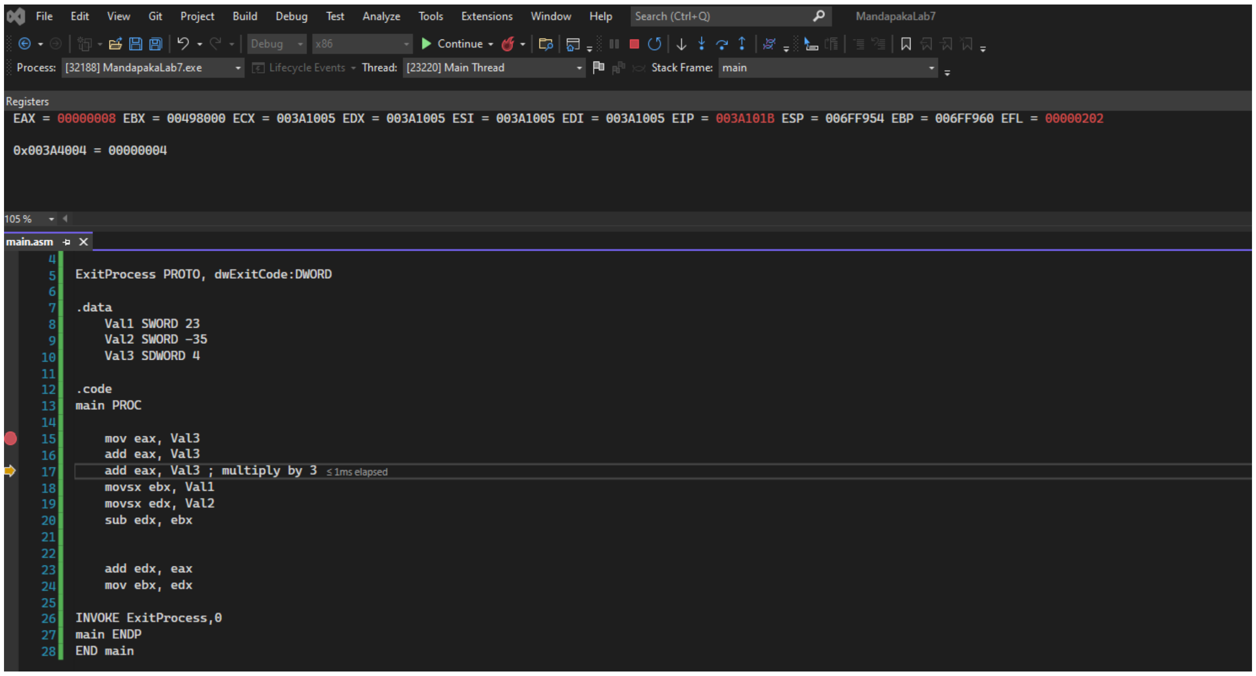
```
 4
 5  ExitProcess PROTO, dwExitCode:DWORD
 6
 7  .data
 8      Val1 SWORD 23
 9      Val2 SWORD -35
10      Val3 SDWORD 4
11
12  .code
13  main PROC
14
15      mov eax, Val3
16      add eax, Val3  ≤1ms elapsed
17      add eax, Val3 ; multiply by 3
18      movsx ebx, Val1
19      movsx edx, Val2
20      sub edx, ebx
21
22
23      add edx, eax
24      mov ebx, edx
25
26  INVOKE ExitProcess,0
27  main ENDP
28  END main
```

Explanation: In the above-provided instruction we are moving the value of Val3, which is'4' into the eax register, and the eax register displays the number at the end and all the numbers before 4 are 0's and we can treat these 0 values as garbage values, these are here to fill up the space.

Line:16

Instruction: add eax, Val 3

Register value: EAX = 00000008



Process: [32188] MandapakaLab7.exe
Thread: [23220] Main Thread

Registers
```
EAX = 00000008 EBX = 00498000 ECX = 003A1005 EDX = 003A1005 ESI = 003A1005 EDI = 003A1005 EIP = 003A101B ESP = 006FF954 EBP = 006FF960 EFL = 00000202

0x003A4004 = 00000004
```

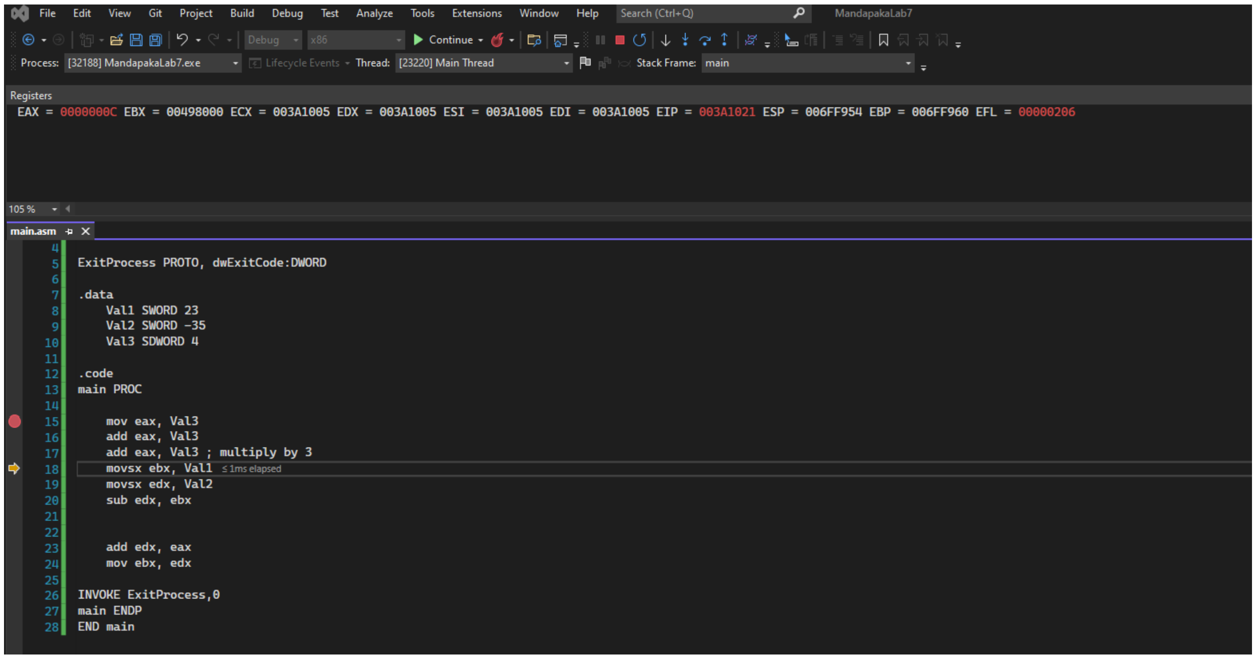
```
 4
 5  ExitProcess PROTO, dwExitCode:DWORD
 6
 7  .data
 8      Val1 SWORD 23
 9      Val2 SWORD -35
10      Val3 SDWORD 4
11
12  .code
13  main PROC
14
15      mov eax, Val3
16      add eax, Val3
17      add eax, Val3 ; multiply by 3  ≤ 1ms elapsed
18      movsx ebx, Val1
19      movsx edx, Val2
20      sub edx, ebx
21
22
23      add edx, eax
24      mov ebx, edx
25
26  INVOKE ExitProcess,0
27  main ENDP
28  END main
```

Explanation: In this instruction we are adding the value that is stored in eax with the initial value in Val3, so from the previous step, we have value 4 in the eax register and now we are adding the value 4 from the variable Val3 to the eax register again, which gives us the value of 8.

Line:17

Instruction: add eax, Val3

Register value: EAX = 0000000C


Registers

```
EAX = 0000000C EBX = 00498000 ECX = 003A1005 EDX = 003A1005 ESI = 003A1005 EDI = 003A1005 EIP = 003A1021 ESP = 006FF954 EBP = 006FF960 EFL = 00000206
```

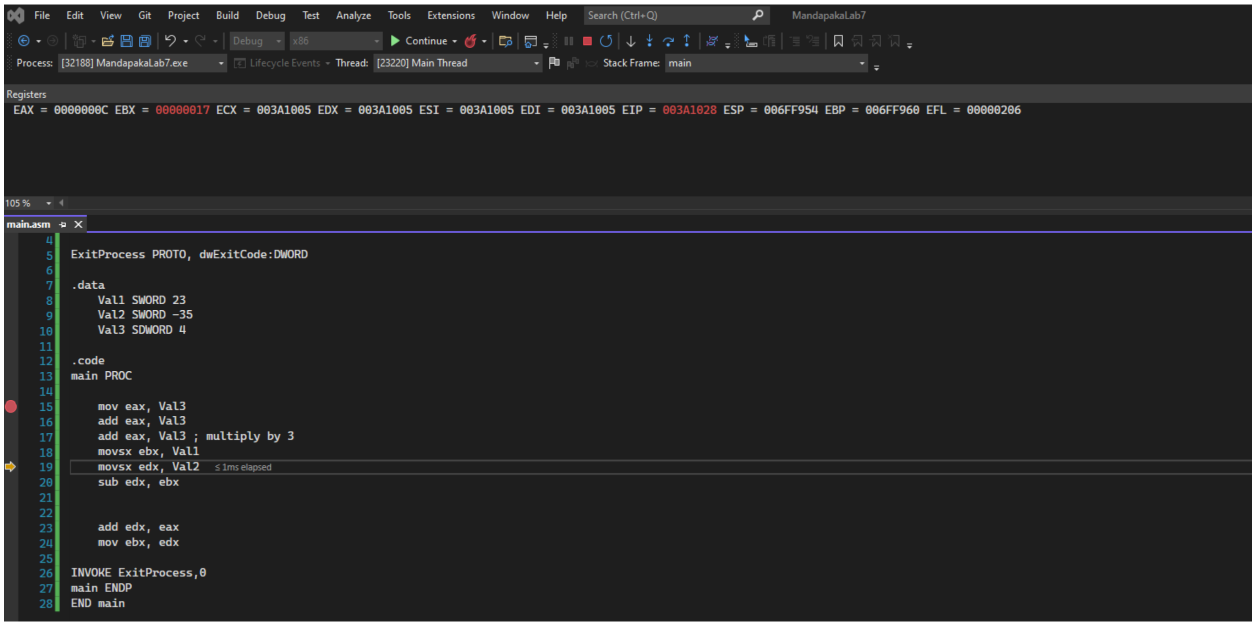
```
ExitProcess PROTO, dwExitCode:DWORD

.data
    Val1 SWORD 23
    Val2 SWORD -35
    Val3 SDWORD 4

.code
main PROC

    mov eax, Val3
    add eax, Val3
    add eax, Val3 ; multiply by 3
    movsx ebx, Val1
    movsx edx, Val2
    sub edx, ebx


    add edx, eax
    mov ebx, edx

INVOKE ExitProcess,0
main ENDP
END main
```

Explanation: Here in the eax register we have the value 8 stored in it from previous steps, so now we are adding the value in Val3 to the current value in the eax register, which is 8+4, which gives us 12, in the eax register it displays the letter'C' , which is basically number 12, but it is C in hexadecimal value. So we can see that the eax register displays the added value which is C also 12.

Line: 18

Instruction: movsx ebx, val1

Register value: EBX = 00000017


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
MandapakaLab7
Debug
x86
Continue
Lifecycle Events
105 %
main.asm

Process: [32188] MandapakaLab7.exe Thread: [23220] Main Thread Stack Frame: main

Registers
```
EAX = 0000000C EBX = 00000017 ECX = 003A1005 EDX = 003A1005 ESI = 003A1005 EDI = 003A1005 EIP = 003A1028 ESP = 006FF954 EBP = 006FF960 EFL = 00000206
```

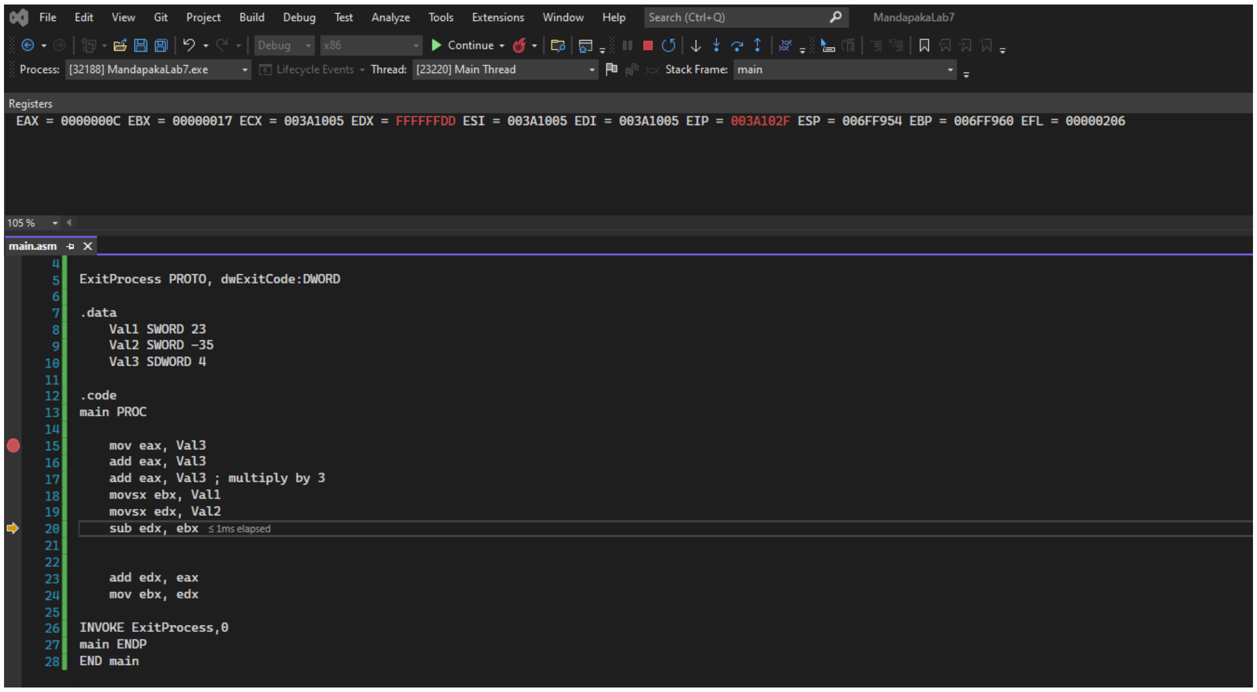
```
 4
 5  ExitProcess PROTO, dwExitCode:DWORD
 6
 7  .data
 8      Val1 SWORD 23
 9      Val2 SWORD -35
10      Val3 SDWORD 4
11
12  .code
13  main PROC
14
15      mov eax, Val3
16      add eax, Val3
17      add eax, Val3 ; multiply by 3
18      movsx ebx, Val1
19      movsx edx, Val2   ≤ 1ms elapsed
20      sub edx, ebx
21
22
23      add edx, eax
24      mov ebx, edx
25
26  INVOKE ExitProcess,0
27  main ENDP
28  END main
```

Explanation: The value in the val1 has been moved to the ebx register.

Line: 19

Instruction: movsx edx, val2

Register value:EDX = FFFFFFDD


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
MandapakaLab7
Debug
x86
Continue
Lifecycle Events
105 %
main.asm

Process: [32188] MandapakaLab7.exe Thread: [23220] Main Thread Stack Frame: main

Registers
```
EAX = 0000000C EBX = 00000017 ECX = 003A1005 EDX = FFFFFFDD ESI = 003A1005 EDI = 003A1005 EIP = 003A102F ESP = 006FF954 EBP = 006FF960 EFL = 00000206
```

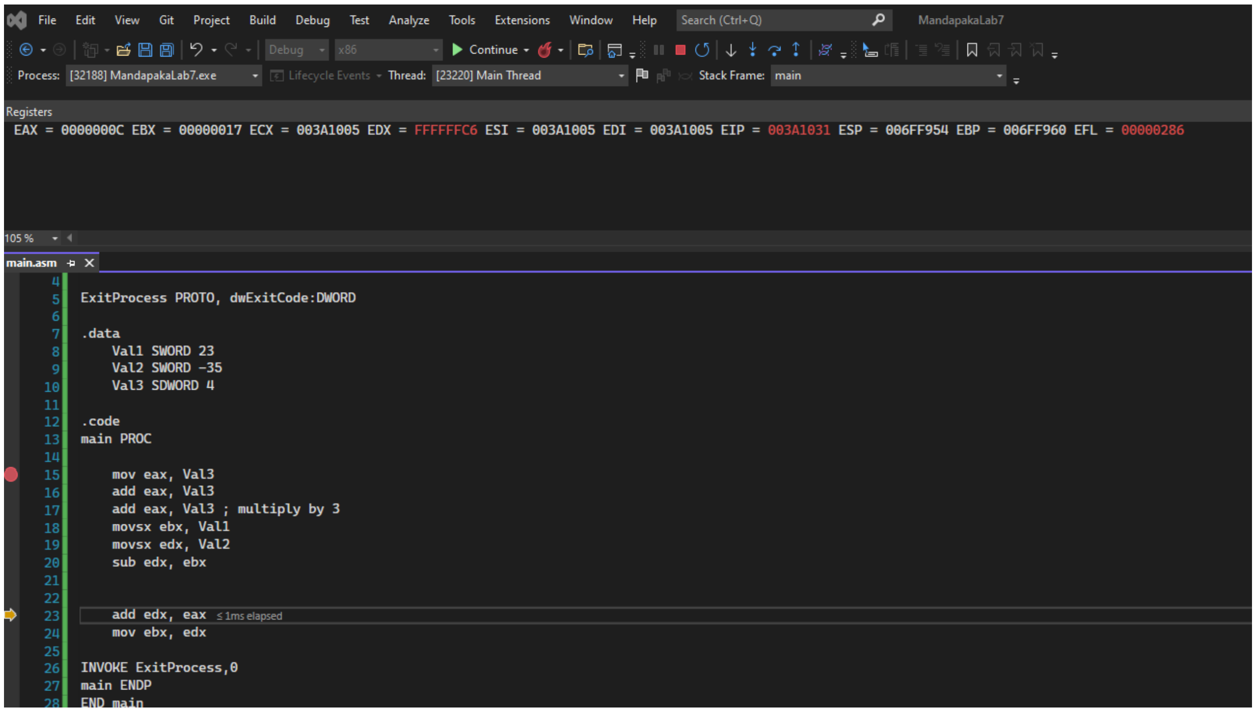
```
 4
 5  ExitProcess PROTO, dwExitCode:DWORD
 6
 7  .data
 8      Val1 SWORD 23
 9      Val2 SWORD -35
10      Val3 SDWORD 4
11
12  .code
13  main PROC
14
15      mov eax, Val3
16      add eax, Val3
17      add eax, Val3 ; multiply by 3
18      movsx ebx, Val1
19      movsx edx, Val2
20      sub edx, ebx   ≤ 1ms elapsed
21
22
23      add edx, eax
24      mov ebx, edx
25
26  INVOKE ExitProcess,0
27  main ENDP
28  END main
```

Explanation: here the value of the val2, which is -35 in decimal has been moved to the edx register and we know that all registers work and give hexadecimal values so the value of val2 is -35, which is FFDD in hexadecimal, so the value FFDD has been stored in the edx register.

Line: 20

Instruction: sub edx, ebx

Register value: EDX = FFFFFFC6


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
MandapakaLab7
Continue
Process: [32188] MandapakaLab7.exe
Thread: [23220] Main Thread
Stack Frame: main
main.asm

Registers
```
EAX = 0000000C EBX = 00000017 ECX = 003A1005 EDX = FFFFFFC6 ESI = 003A1005 EDI = 003A1005 EIP = 003A1031 ESP = 006FF954 EBP = 006FF960 EFL = 00000286
```

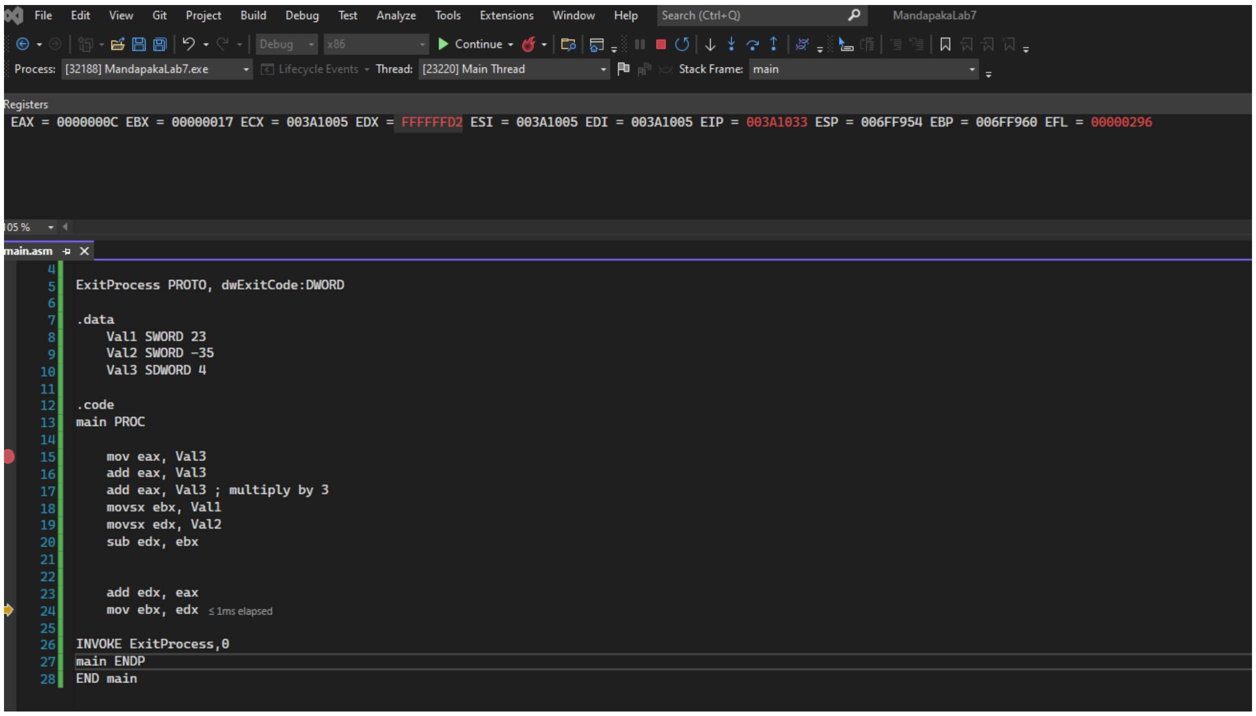
```
ExitProcess PROTO, dwExitCode:DWORD

.data
    Val1 SWORD 23
    Val2 SWORD -35
    Val3 SDWORD 4

.code
main PROC

    mov eax, Val3
    add eax, Val3
    add eax, Val3 ; multiply by 3
    movsx ebx, Val1
    movsx edx, Val2
    sub edx, ebx


    add edx, eax  ≤ 1ms elapsed
    mov ebx, edx

INVOKE ExitProcess,0
main ENDP
END main
```

Explanation: Here I am subtracting the value from the edx register with the ebx register which is subtracting val2 with val1 and the sum or the answer of that subtraction will be stored in the edx register

Line: 23

Instruction: add edx, eax

Register value: EDX = FFFFFFD2



Process: [32188] MandapakaLab7.exe Thread: [23220] Main Thread

```
Registers
EAX = 0000000C EBX = 00000017 ECX = 003A1005 EDX = FFFFFFD2 ESI = 003A1005 EDI = 003A1005 EIP = 003A1033 ESP = 006FF954 EBP = 006FF960 EFL = 00000296
```

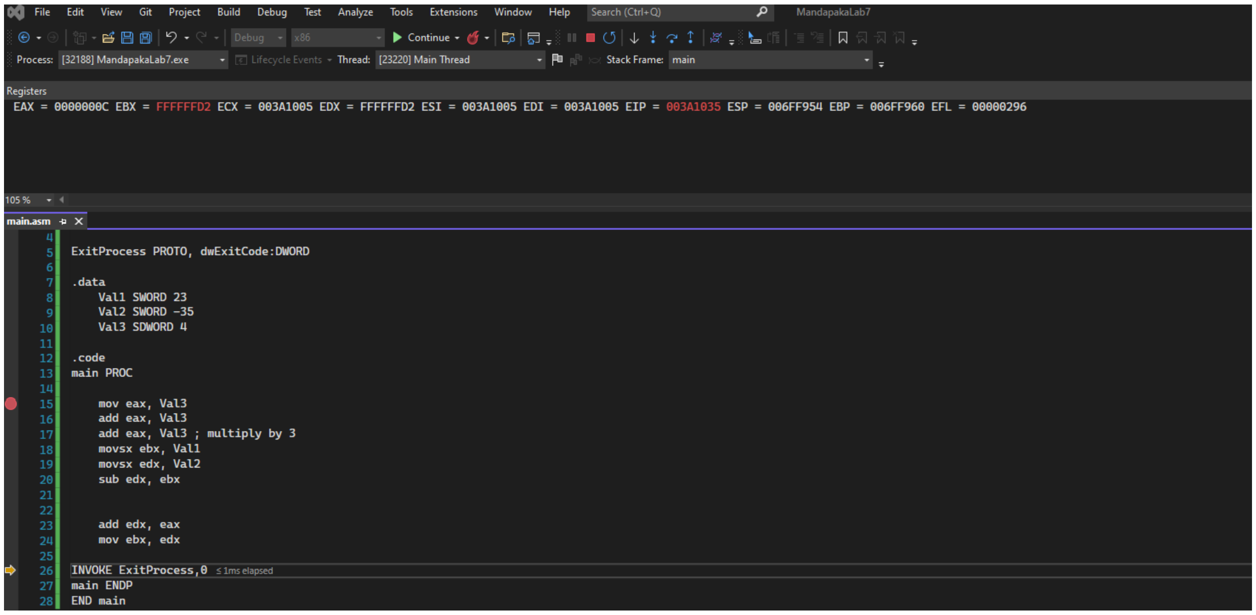
```
4
5   ExitProcess PROTO, dwExitCode:DWORD
6
7   .data
8       Val1 SWORD 23
9       Val2 SWORD -35
10      Val3 SDWORD 4
11
12  .code
13  main PROC
14
15      mov eax, Val3
16      add eax, Val3
17      add eax, Val3 ; multiply by 3
18      movsx ebx, Val1
19      movsx edx, Val2
20      sub edx, ebx
21
22
23      add edx, eax
24      mov ebx, edx  ≤ 1ms elapsed
25
26  INVOKE ExitProcess,0
27  main ENDP
28  END main
```

Explanation: in this instruction we are adding the value of edx, which is -58 with the value in ebx which is 12 which gives us -48 and the hexadecimal value for -48 is D2.

Line: 24

Instruction: mov ebx, edx

Register value: EBX = FFFFFFD2

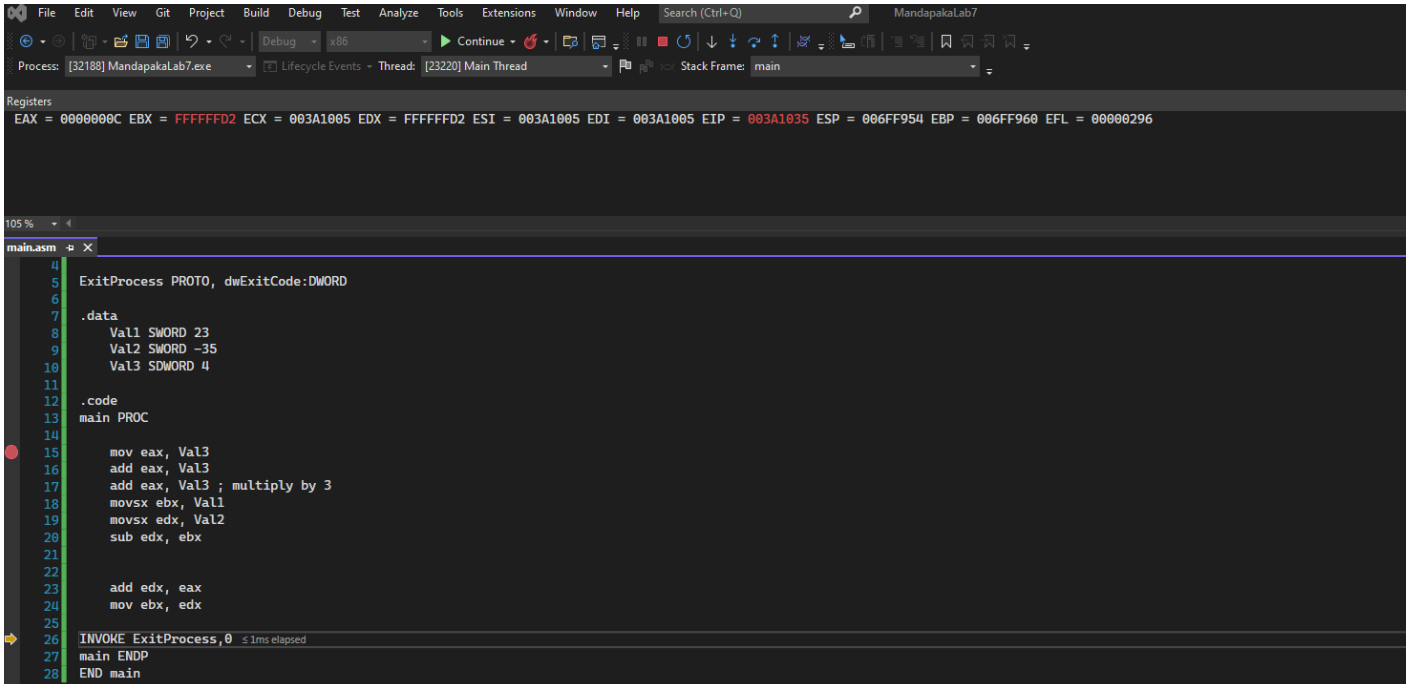


```
EAX = 0000000C EBX = FFFFFFD2 ECX = 003A1005 EDX = FFFFFFD2 ESI = 003A1005 EDI = 003A1005 EIP = 003A1035 ESP = 006FF954 EBP = 006FF960 EFL = 00000296
```

```
ExitProcess PROTO, dwExitCode:DWORD

.data
    Val1 SWORD 23
    Val2 SWORD -35
    Val3 SDWORD 4

.code
main PROC

    mov eax, Val3
    add eax, Val3
    add eax, Val3 ; multiply by 3
    movsx ebx, Val1
    movsx edx, Val2
    sub edx, ebx


    add edx, eax
    mov ebx, edx

INVOKE ExitProcess,0
main ENDP
END main
```

Explanation: Here I have performed the operation in the edx register and I did mov operation because I want to move the sum value of the evaluated expression into the ebx register.

Complete for the all the lines inside main


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
MandapakaLab7
Continue
Process: [32188] MandapakaLab7.exe
Thread: [23220] Main Thread
Stack Frame: main
Registers
105 %
main.asm
≤ 1ms elapsed

```
EAX = 0000000C EBX = FFFFFFD2 ECX = 003A1005 EDX = FFFFFFD2 ESI = 003A1005 EDI = 003A1005 EIP = 003A1035 ESP = 006FF954 EBP = 006FF960 EFL = 00000296
```

```
ExitProcess PROTO, dwExitCode:DWORD

.data
    Val1 SWORD 23
    Val2 SWORD -35
    Val3 SDWORD 4

.code
main PROC

    mov eax, Val3
    add eax, Val3
    add eax, Val3 ; multiply by 3
    movsx ebx, Val1
    movsx edx, Val2
    sub edx, ebx


    add edx, eax
    mov ebx, edx

INVOKE ExitProcess,0
main ENDP
END main
```