COAL LAB 4 - 24K0546

Task 1

INCLUDE Irvine32.inc

.data

    num1 BYTE 25 ; equivalent to 19h
    num2 WORD 01200h
    num3 DWORD ?

.code

    main PROC

        MOV AL, num1 ; AL becomes 19h
        call DumpRegs

        MOV AX, num2 ; AX becomes 1200h
        call DumpRegs

        MOVZX EAX, AX ; EAX becomes 1200h with zero-extends
        MOV num3, EAX ; MOV EAX to num3
        call DumpRegs

    exit
    main ENDP
    END main

Task 2

```
INCLUDE Irvine32.inc
.data
    var1 SBYTE -50 ; CE in SBYTE hexadecimal representation - FFFF FFCE in DWORD
    var2 BYTE 200  ; C8 in BYTE hexadecimal representation
.code
    main PROC

        MOVSX EAX, var1 ; MOV var1 with sign-extend
        call DumpRegs

        MOVZX EAX, var2 ; MOV var2 with zero-extend
        call DumpRegs

    exit
    main ENDP
    END main
```


Task 3

```
INCLUDE Irvine32.inc
.data
    num1 WORD 01234h
    num2 WORD 05678h
.code
    main PROC
        MOVZX EAX, num1 ; Initial num1
        call DumpRegs
```

MOV AX, num2 ; Initial num2

call DumpRegs

MOV AX, num1

XCHG AX, num2

MOV num1, AX

MOV AX, num1 ; num1 swapped value

call DumpRegs

MOV AX, num2 ; num2 swapped value

call DumpRegs

MOV AX, num1

XCHG AX, num2

MOV num1, AX

MOV AX, num1 ; num1 reswapped value

call DumpRegs

MOV AX, num2 ; num2 reswapped value

call DumpRegs

exit

main ENDP

END main

Task 4

```asm
INCLUDE Irvine32.inc
.data
    Xval DWORD 25
    Yval DWORD 15
    Zval DWORD 40
    Rval DWORD ?
.code
  main PROC
      MOV EAX, Xval
      ADD EAX, Yval
      SUB EAX, Zval
      NEG EAX
      MOV Rval, EAX
      call WriteInt
  exit
  main ENDP
  END main
```
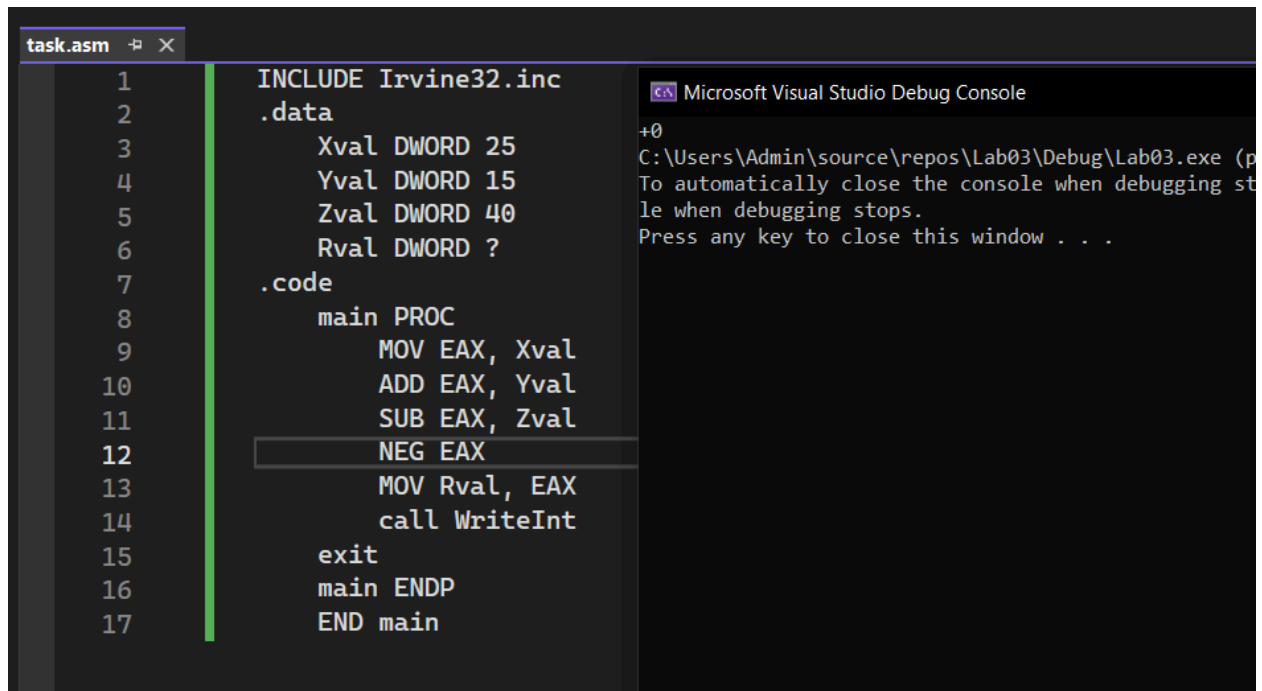
Task 5
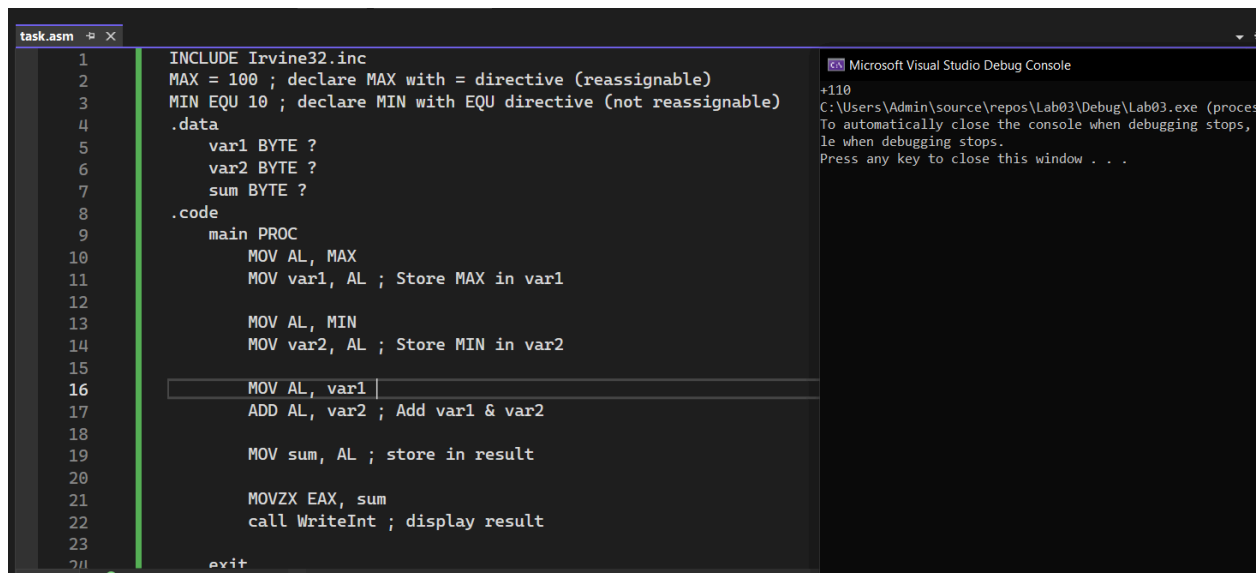
```asm
INCLUDE Irvine32.inc
MAX = 100 ; declare MAX with = directive (reassignable)
MIN EQU 10 ; declare MIN with EQU directive (not reassignable)
.data
    var1 BYTE ?
    var2 BYTE ?
    sum BYTE ?
.code
    main PROC
        MOV AL, MAX
        MOV var1, AL ; Store MAX in var1
        MOV AL, MIN
        MOV var2, AL ; Store MIN in var2
        MOV AL, var1
        ADD AL, var2 ; Add var1 & var2
        MOV sum, AL ; store in result
        MOVZX EAX, sum
        call WriteInt ; display result
    exit
    main ENDP
    END main
```