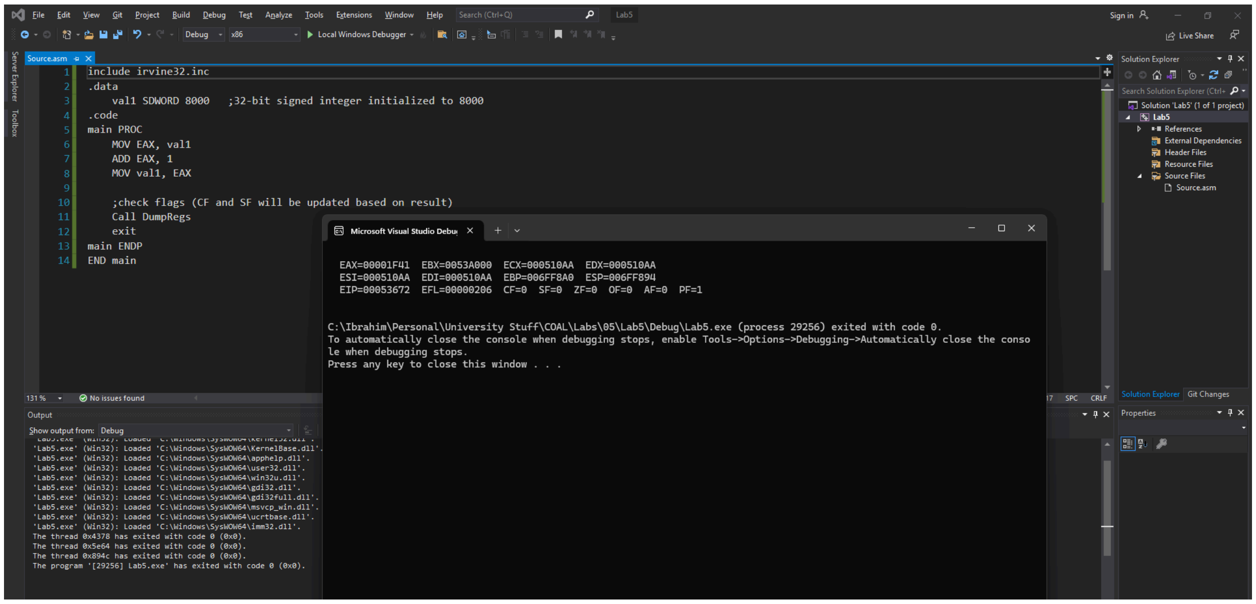
COAL LAB 05 - 06 Mar 2025

Student Name: Ibrahim Johar Farooqi

Student ID: 23K-0074

Task 1:



```
include irvine32.inc
.data
    val1 SDWORD 8000   ;32-bit signed integer initialized to 8000
.code
main PROC
    MOV EAX, val1
    ADD EAX, 1
    MOV val1, EAX

    ;check flags (CF and SF will be updated based on result)
    Call DumpRegs
    exit
main ENDP
END main
```

```
  EAX=00001F41  EBX=0053A000  ECX=000510AA  EDX=000510AA
  ESI=000510AA  EDI=000510AA  EBP=006FF8A0  ESP=006FF894
  EIP=00053672  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


C:\Ibrahim\Personal\University Stuff\COAL\Labs\05\Lab5\Debug\Lab5.exe (process 29256) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

```
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\KernelBase.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\apphelp.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\user32.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\win32u.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\gdi32.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\gdi32full.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\msvcp_win.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ucrtbase.dll'.
'Lab5.exe' (Win32): Loaded 'C:\Windows\SysWOW64\imm32.dll'.
The thread 0x4378 has exited with code 0 (0x0).
The thread 0x5e64 has exited with code 0 (0x0).
The thread 0x894c has exited with code 0 (0x0).
The program '[29256] Lab5.exe' has exited with code 0 (0x0).
```

- Carry Flag (CF) = 0 (no unsigned overflow)
- Sign Flag (SF) = 0 (result is positive)

Task 2:

MOV AX, 7FF0h

ADD AL, 10h

a. CF = 1, SF = 0, ZF = 1, OF = 0

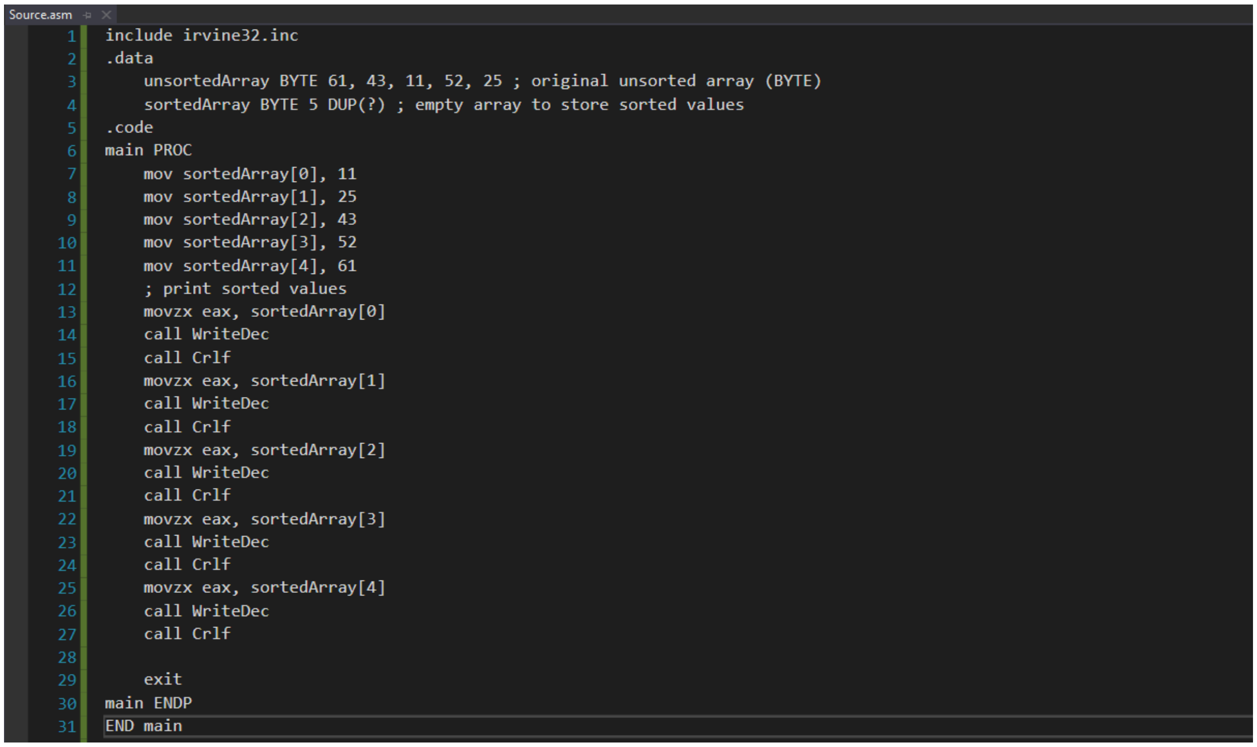
ADD AH, 1

b. CF = 0, SF = 1, ZF = 0, OF = 1

ADD AX, 2

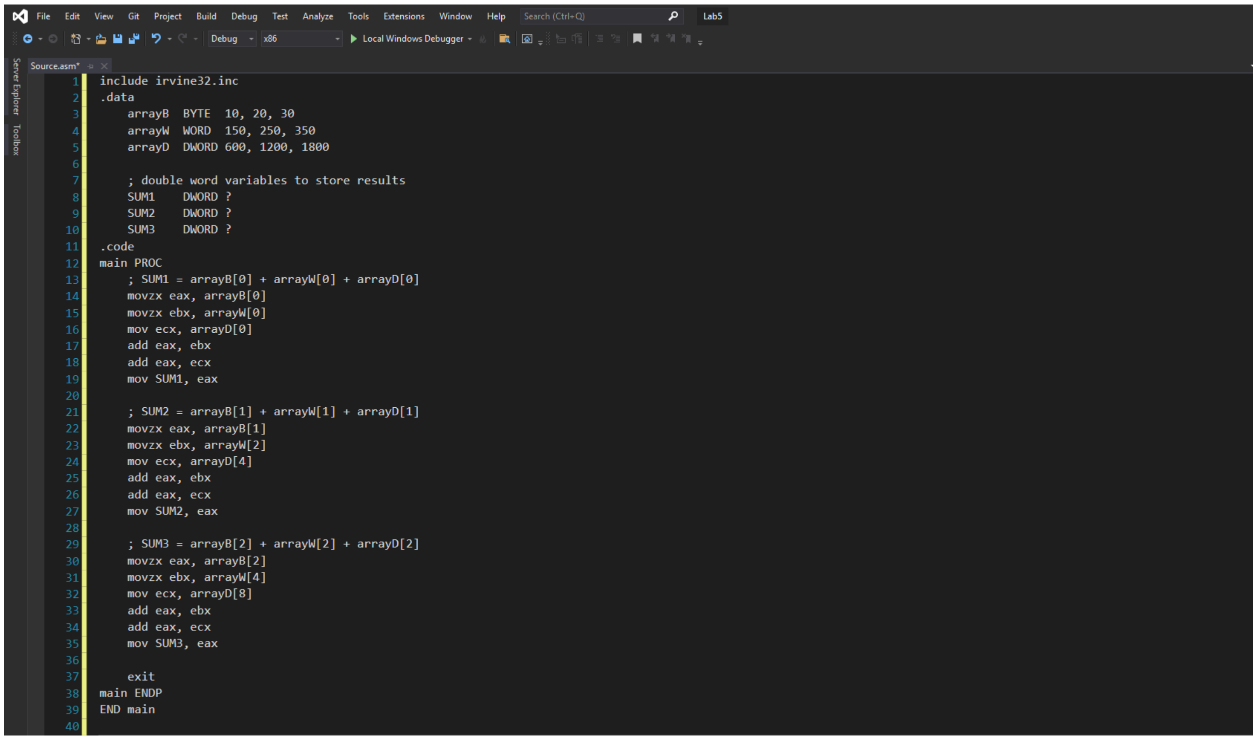
c. CF = 0, SF = 1, ZF = 0, OF = 0

Task 3:



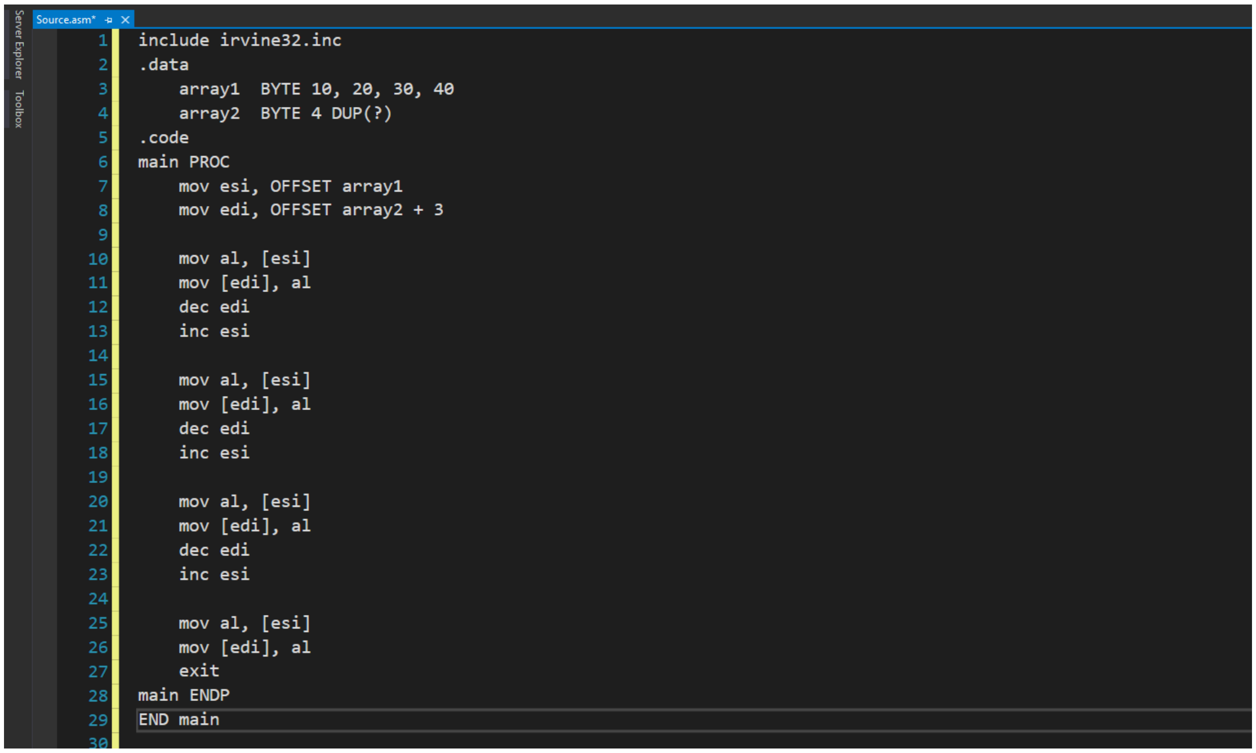
```
include irvine32.inc
.data
    unsortedArray BYTE 61, 43, 11, 52, 25 ; original unsorted array (BYTE)
    sortedArray BYTE 5 DUP(?) ; empty array to store sorted values
.code
main PROC
    mov sortedArray[0], 11
    mov sortedArray[1], 25
    mov sortedArray[2], 43
    mov sortedArray[3], 52
    mov sortedArray[4], 61
    ; print sorted values
    movzx eax, sortedArray[0]
    call WriteDec
    call Crlf
    movzx eax, sortedArray[1]
    call WriteDec
    call Crlf
    movzx eax, sortedArray[2]
    call WriteDec
    call Crlf
    movzx eax, sortedArray[3]
    call WriteDec
    call Crlf
    movzx eax, sortedArray[4]
    call WriteDec
    call Crlf

    exit
main ENDP
END main
```

Task 4:



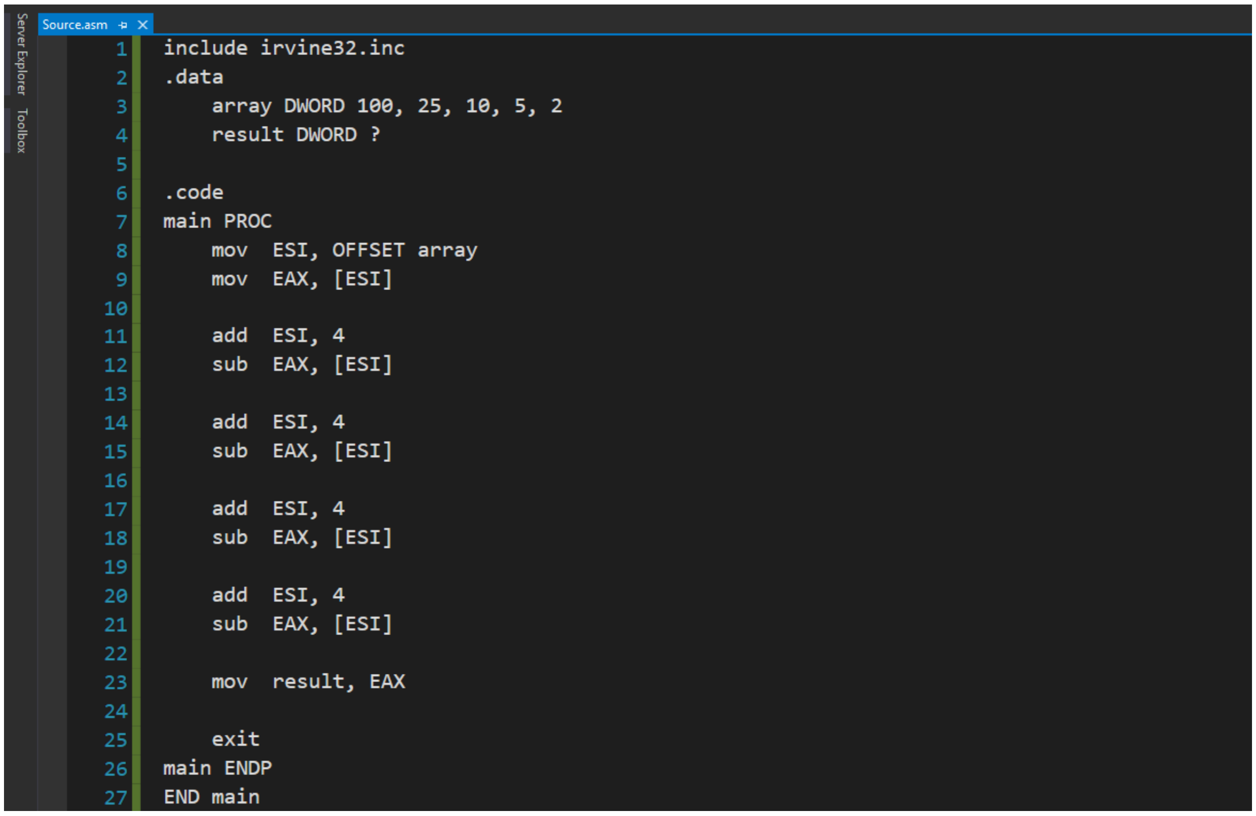
```
include irvine32.inc
.data
    arrayB  BYTE  10, 20, 30
    arrayW  WORD  150, 250, 350
    arrayD  DWORD 600, 1200, 1800

    ; double word variables to store results
    SUM1    DWORD ?
    SUM2    DWORD ?
    SUM3    DWORD ?
.code
main PROC
    ; SUM1 = arrayB[0] + arrayW[0] + arrayD[0]
    movzx eax, arrayB[0]
    movzx ebx, arrayW[0]
    mov ecx, arrayD[0]
    add eax, ebx
    add eax, ecx
    mov SUM1, eax

    ; SUM2 = arrayB[1] + arrayW[1] + arrayD[1]
    movzx eax, arrayB[1]
    movzx ebx, arrayW[2]
    mov ecx, arrayD[4]
    add eax, ebx
    add eax, ecx
    mov SUM2, eax

    ; SUM3 = arrayB[2] + arrayW[2] + arrayD[2]
    movzx eax, arrayB[2]
    movzx ebx, arrayW[4]
    mov ecx, arrayD[8]
    add eax, ebx
    add eax, ecx
    mov SUM3, eax

    exit
main ENDP
END main
```

Task 5:



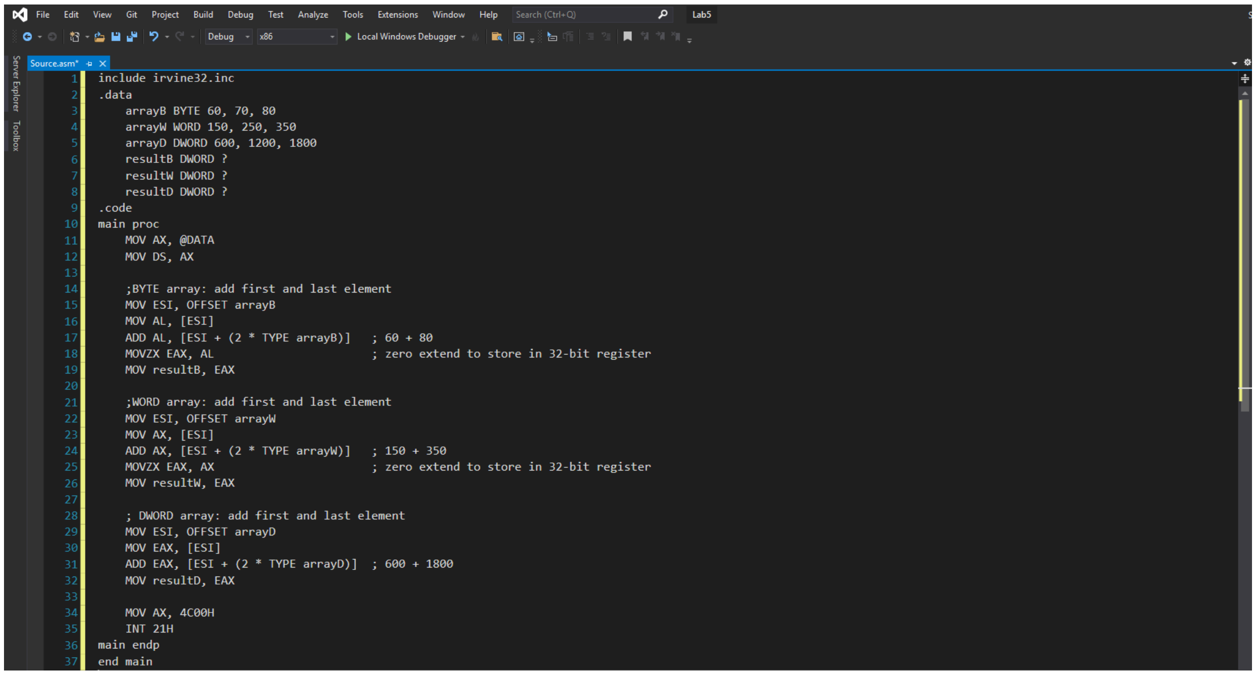
```
include irvine32.inc
.data
    array1  BYTE 10, 20, 30, 40
    array2  BYTE 4 DUP(?)
.code
main PROC
    mov esi, OFFSET array1
    mov edi, OFFSET array2 + 3

    mov al, [esi]
    mov [edi], al
    dec edi
    inc esi

    mov al, [esi]
    mov [edi], al
    dec edi
    inc esi

    mov al, [esi]
    mov [edi], al
    dec edi
    inc esi

    mov al, [esi]
    mov [edi], al
    exit
main ENDP
END main
```

Task 6:



```
include irvine32.inc
.data
    array DWORD 100, 25, 10, 5, 2
    result DWORD ?

.code
main PROC
    mov  ESI, OFFSET array
    mov  EAX, [ESI]

    add  ESI, 4
    sub  EAX, [ESI]

    add  ESI, 4
    sub  EAX, [ESI]

    add  ESI, 4
    sub  EAX, [ESI]

    add  ESI, 4
    sub  EAX, [ESI]

    mov  result, EAX

    exit
main ENDP
END main
```

Task 7:



```
include irvine32.inc
.data
    arrayB BYTE 60, 70, 80
    arrayW WORD 150, 250, 350
    arrayD DWORD 600, 1200, 1800
    resultB DWORD ?
    resultW DWORD ?
    resultD DWORD ?
.code
main proc
    MOV AX, @DATA
    MOV DS, AX

    ;BYTE array: add first and last element
    MOV ESI, OFFSET arrayB
    MOV AL, [ESI]
    ADD AL, [ESI + (2 * TYPE arrayB)]   ; 60 + 80
    MOVZX EAX, AL                       ; zero extend to store in 32-bit register
    MOV resultB, EAX

    ;WORD array: add first and last element
    MOV ESI, OFFSET arrayW
    MOV AX, [ESI]
    ADD AX, [ESI + (2 * TYPE arrayW)]   ; 150 + 350
    MOVZX EAX, AX                       ; zero extend to store in 32-bit register
    MOV resultW, EAX

    ; DWORD array: add first and last element
    MOV ESI, OFFSET arrayD
    MOV EAX, [ESI]
    ADD EAX, [ESI + (2 * TYPE arrayD)]  ; 600 + 1800
    MOV resultD, EAX

    MOV AX, 4C00H
    INT 21H
main endp
end main
```