Iqra University Islamabad

Name: Aksah Akhalq

Student ID: 41593

Submitted To: Sir Asad Ullah Khan

Submission Date: November 20, 2025

Department: Computer Science

LAB TASK 5

```
TITLE Memory Operations Lab (memory_ops.asm)
; Program Description: This program demonstrates storing,
incrementing, and adding
; bytes in memory.
.686
.MODEL FLAT, STDCALL
.STACK
INCLUDE Irvine32.inc
.DATA
; Task 1: Ten bytes initialized to 1
byteArray1 BYTE 10, 20, 30, 40, 50, 60, 70, 80, 90, 0Ah
; Task 2: Sixteen bytes initialized to 0
byteArray2 BYTE 16 DUP(0)
; Task 3: Variable to store sum
sum DWORD 0
.CODE
main PROC
; Task 2: Increment each byte in byteArray2 by 1
mov esi, OFFSET byteArray2
mov ecx, 16
L1:
L2:
inc BYTE PTR [esi]
inc esi
loop L1
; Task 3: Sum ten bytes from byteArray1
mov esi, OFFSET byteArray1
mov ecx, 10
mov eax, 0
```

movzx ebx, BYTE PTR [esi]

add eax, ebx

inc esi

loop L2

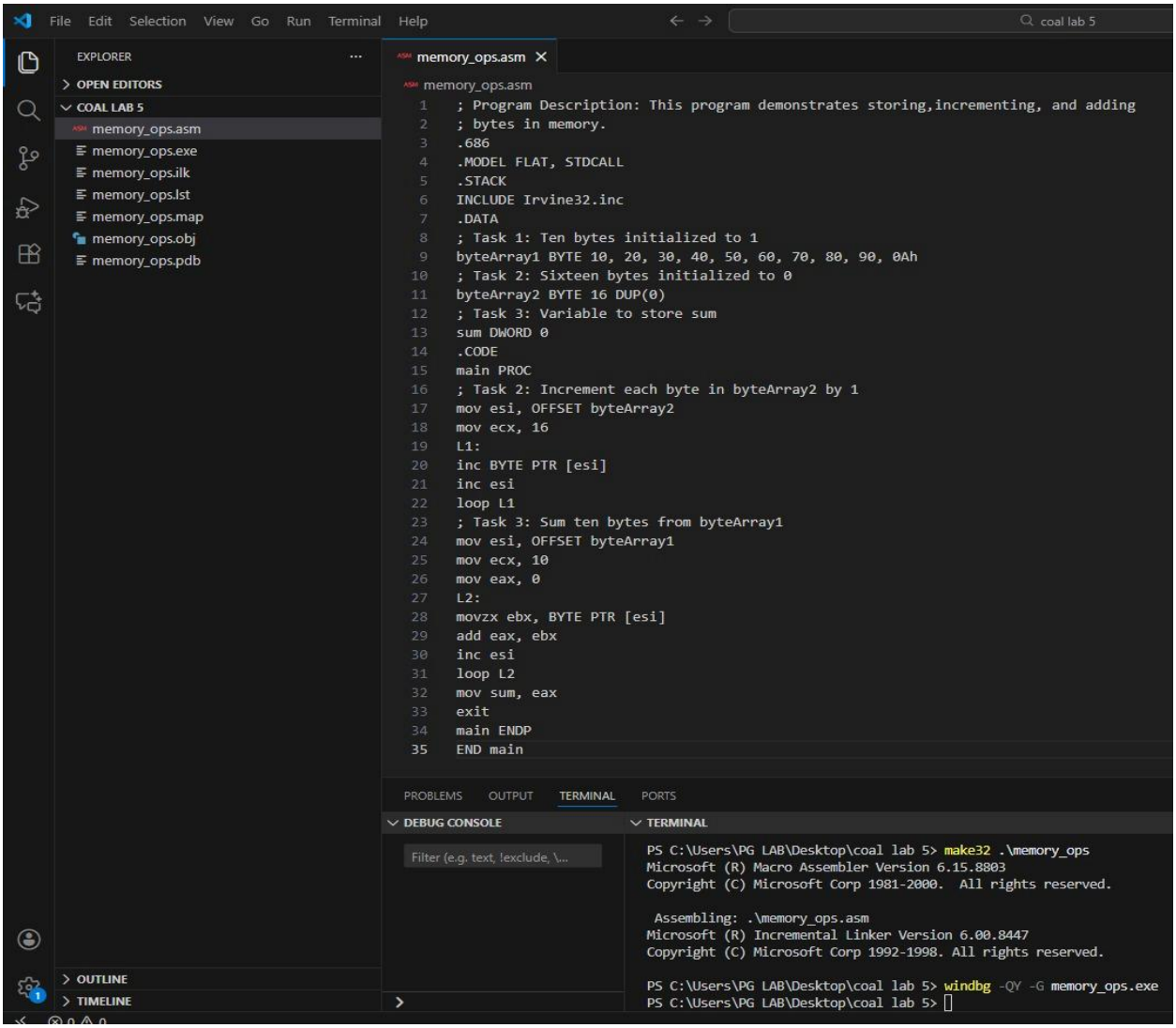
mov sum, eax

exit

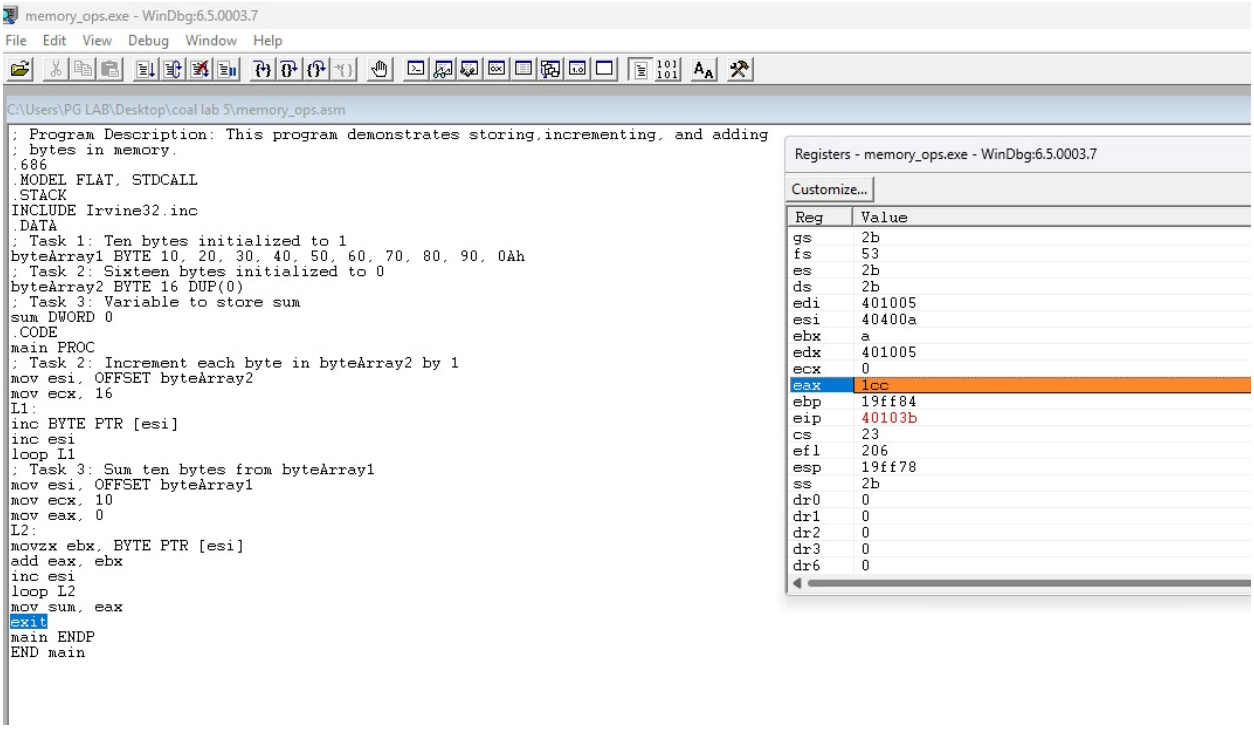
main ENDP

END main

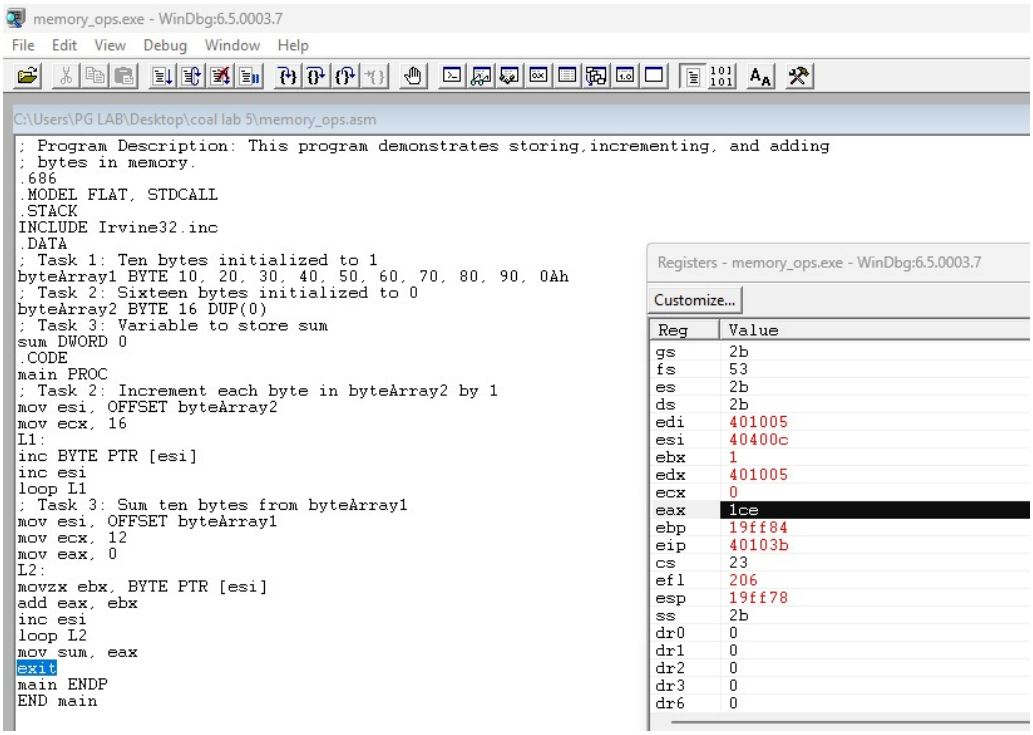
Terminal Output



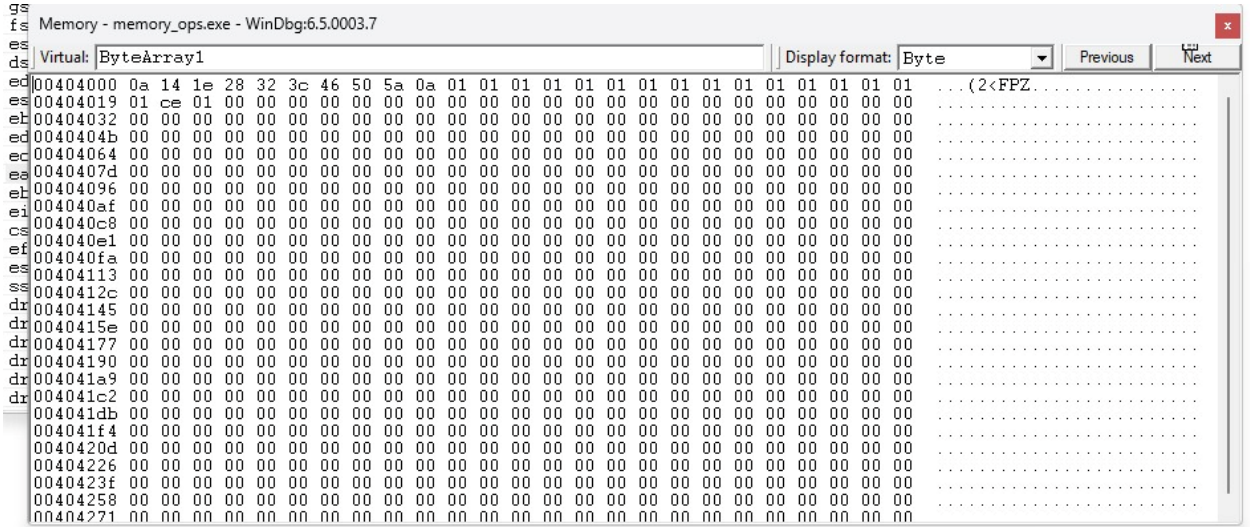
Debugger output



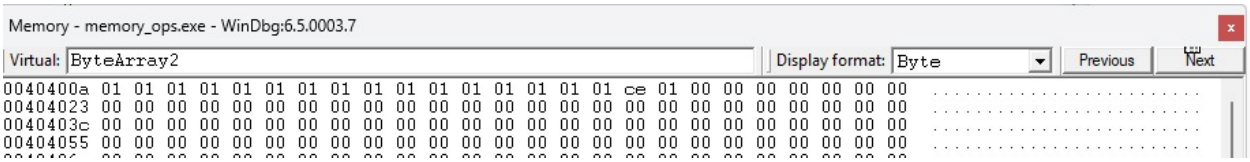
Output For Changed Value



Memory Output (Byte arr1)



Memory Output (Byte arr2)



Q1. Purpose of ESI and EDI

- **ESI:** Used as a *source index* to read bytes from memory (arrays).
- **EDI:** Used as a *destination index* to write bytes to memory (not used in this code).

Q2. LOOP instruction

- Automatically **decrements ECX** and **jumps** if ECX ≠ 0.
- Makes repeating through memory simpler and avoids manual decrement + jump instructions.

Q3. MOV AL, [esi] vs MOV [edi], AL

- **MOV AL, [esi]** → Reads a byte *from memory into AL*.
- **MOV [edi], AL** → Writes a byte *from AL into memory*.

Q4. What if the sum exceeds 8-bit limit?

- An 8-bit register would **overflow** (wrap around after 255).
- To avoid this, use a **32-bit register (EAX)** or check for carry (CF) after each add.

Q5. Modified code: sum only even-indexed bytes

```
mov esi, OFFSET byteArray1
mov ecx, 5
mov eax, 0

SumEven:
   movzx ebx, BYTE PTR [esi]
   add eax, ebx
   add esi, 2          ; go to next even index
loop SumEven

mov sum, eax
```