**Task 1**

“=” only stores integers vales

“equ” stores both integers and strings

“textequ” only stores constant string values

**Task 2**

.386

.model flat,stdcall

.stack 4096

.data

arr dw 1,2,3,4,5,6,7,8,9,0

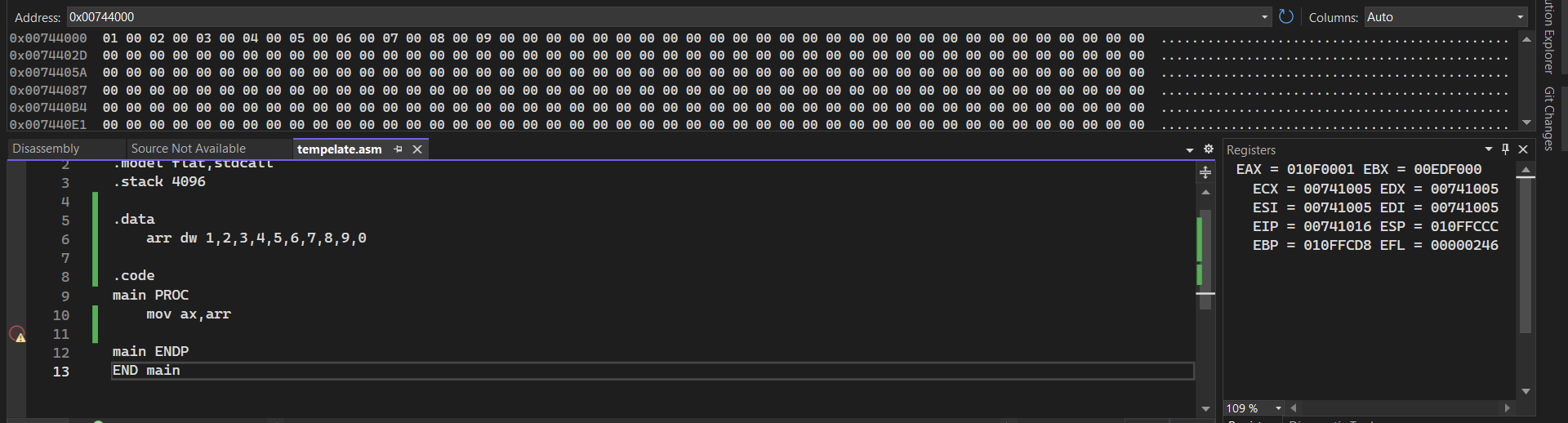
.code

main PROC

mov ax,arr

main ENDP

END main



**Task 3 part(a)**

.386

.model flat,stdcall

.stack 4096

.data

arr1 db 1,2,3,0,5,7

arr1\_0 db 3 ;index for byte

arr2 dw 2,5,0,4,3,2

arr2\_0 dw 2 ;index for word

arr3 dd 11,15,13,0,17

arr3\_0 dd 5 ;index for dword

.code

main PROC

mov al,arr1

mov al,arr1+1

mov al,arr1+2

mov al,arr1+3

mov al,arr1+4

mov al,arr1+5

mov ax,0

mov ax,arr2

mov ax,arr2+2

mov ax,arr2+4

mov ax,arr2+6

mov ax,arr2+8

mov ax,arr2+10

mov ax,0

mov eax,arr3

mov eax,arr3+4

mov eax,arr3+8

mov eax,arr3+12

mov eax,arr3+16

mov eax,arr3+20

mov eax,sizeof arr1

mov eax,sizeof arr2

mov eax,sizeof arr3

mov ax,0

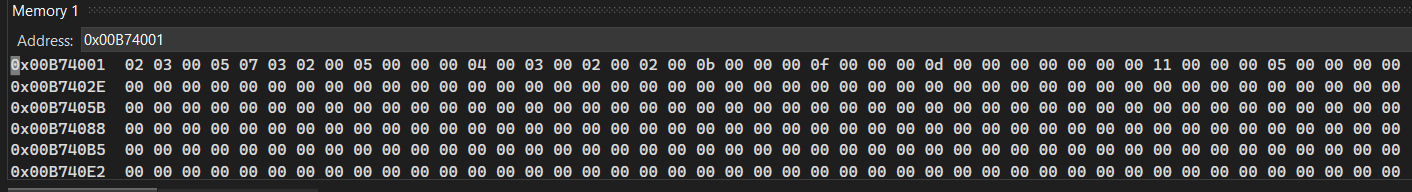
main ENDP

END main

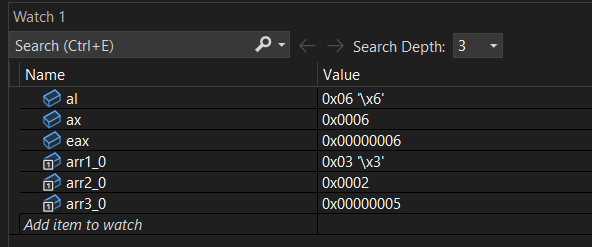
**Byte type** first 6 elements are byte as they take 1 byte space and we need to jump 1 space.

**Word Type** 6 elements of arr2 are word as they take 2 byte space and we need to jump every 2 block to access that memory space.

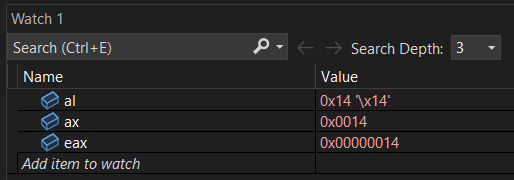
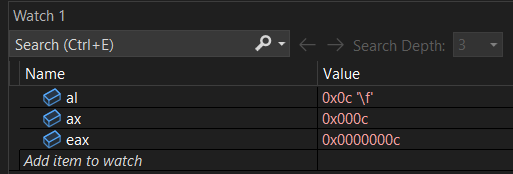
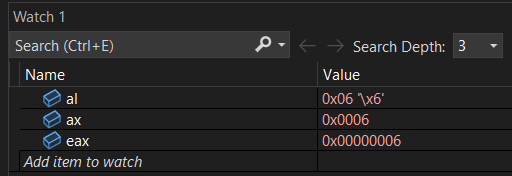
**Dword Type** 6 elements of arr3 are Dword as they take 24byte space and we need to jump every 2 block to access that memory space.

****

**Task 3 part(b)**

****

**Task 3 part(c)**

****

**Task 3 part(d)**

.386

.model flat,stdcall

.stack 4096

.data

arr dw 10h, 20h, 30h, 40h, 50h

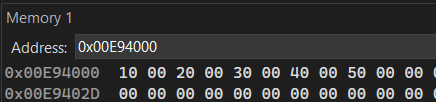
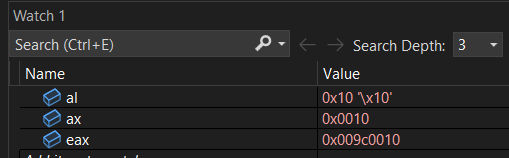
.code

main PROC

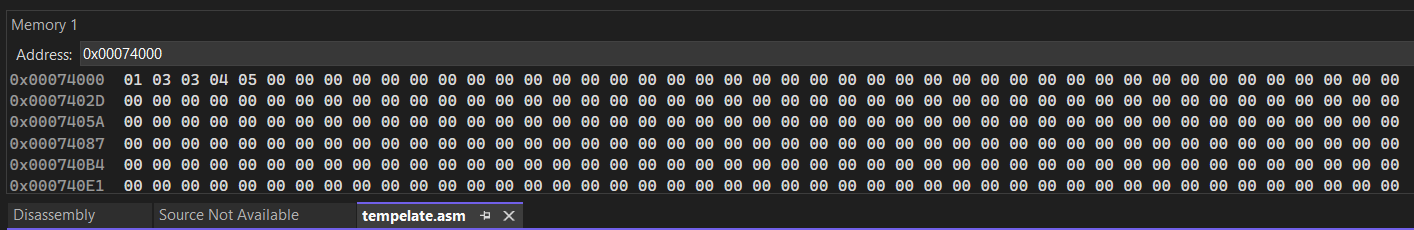
mov ax,word ptr arr

main ENDP

END main

**** ****

**Task 4**

****

.386

.model flat,stdcall

.stack 4096

.data

arr db 1,2,3,4,5

.code

main PROC

mov bl,arr+1

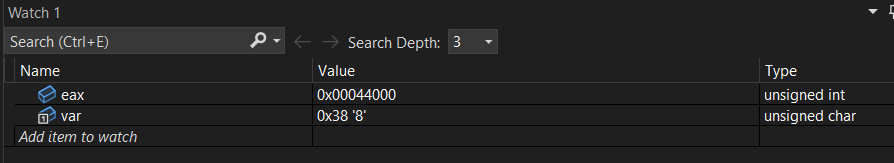
add bl,1

mov [arr+1],bl

main ENDP

END main

**Task 5**



.386

.model flat,stdcall

.stack 4096

.data

var db 56

.code

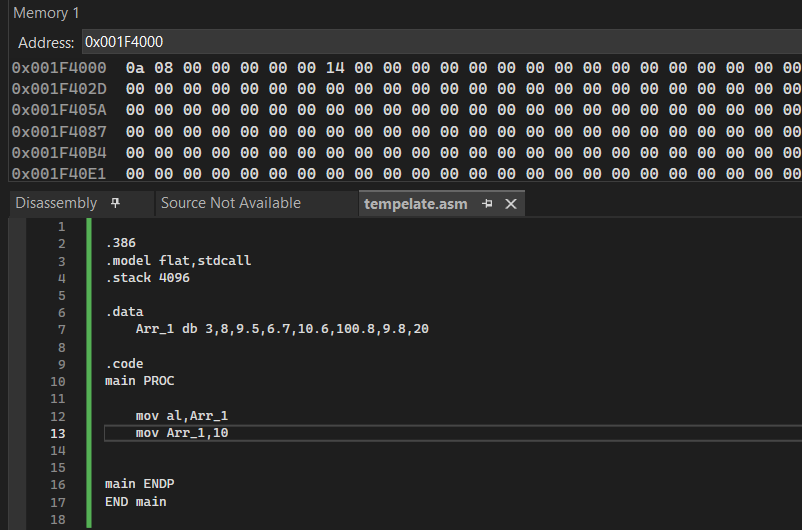
main PROC

mov eax,offset var

main ENDP

END main

**Task 6**

****

.386

.model flat,stdcall

.stack 4096

.data

Arr\_1 db 3,8,9.5,6.7,10.6,100.8,9.8,20

.code

main PROC

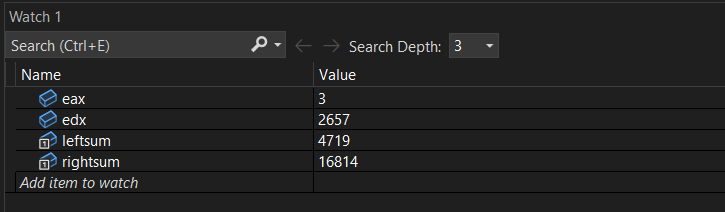
mov al,Arr\_1

mov Arr\_1,10

main ENDP

END main

**Task 7**

****

.386

.model flat,stdcall

.stack 4096

.data

arr dword 16,8,14,4,7,19

var1 dword 1000

var2 dword 100

var3 dword 1

rightsum dword 0

leftsum dword 0

.code

main PROC

mov eax,arr

mul var1

add rightsum,eax

mov eax,arr+4

mul var2

add rightsum,eax

mov eax,arr+8

mul var3

add rightsum,eax

mov eax,arr+12

mul var1

add leftsum,eax

mov eax,arr+16

mul var2

add leftsum,eax

mov eax,arr+20

mul var3

add leftsum,eax

mov edx,0

mov eax,rightsum

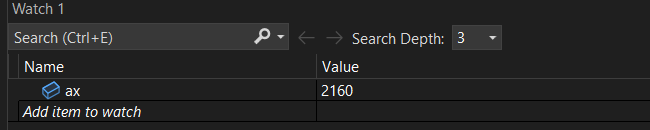
mov ecx,leftsum

div ecx

main ENDP

END main

**Task 8**



.386

.model flat, stdcall

.stack 4096

.data

arr DD 12345678h

var DB ?

var2 byte 12h

.code

main PROC

mov EAX, arr

mov var, AL

mov AX,0

mov BL,var

mov AL,var2

MUL BL

main ENDP

end main