**LAB 6**

**Activity 1**

**.data**

**arrayW WORD 1000h,3000h,4000h**

**arrayD DWORD 1,2,3,4,9**

**; will the following assemble and run?**

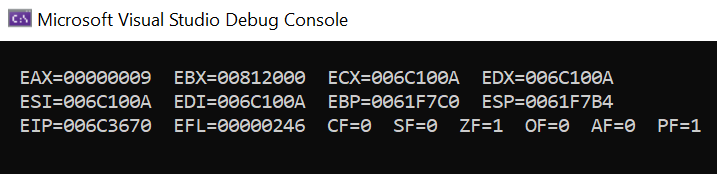
**.code**

**mov ax,[arrayW-2] ; ??**

**mov eax,[arrayD+16] ; ??**

**Ans.** Assembles and runs fine. The first line in .code gives garbage or zero value in ax, while the second line stores the array element 9 in eax register.

**Output:**



**Steps:**





**Activity 2**

Use following array declarations:

arrayB BYTE 5, 4, 2

arrayW WORD 15, 25, 10

arrayD DWORD 60, 12, 18

Now initialize three double word variables SUM1, SUM2, SUM3 and perform

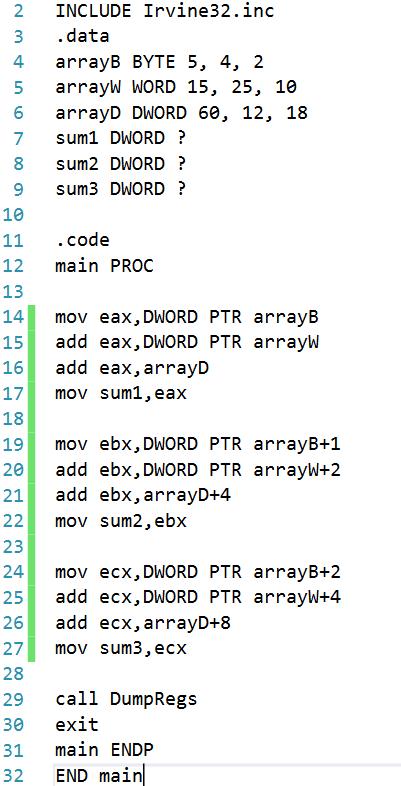
following operations (expressed in pseudo-code here):

SUM1 = arrayB[0]+arrayW[0]+arrayD[0]

SUM2 =arrayB[1]+arrayW[1]+arrayD[1]

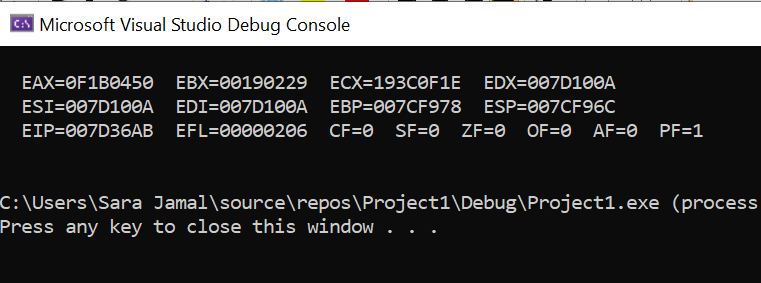
SUM3 =arrayB[2] + arrayW[2] +arrayD[2]

**Program:**

****

; K190207 Sara Jamal sec C

**Output:**



**Activity 3**

**Use following array declarations:**

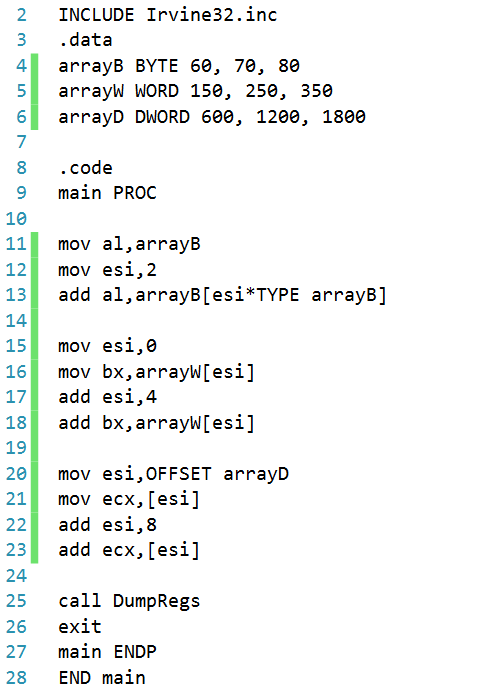
**arrayB BYTE 60, 90, 80**

**arrayW WORD 150, 250, 350**

**arrayD DWORD 600, 1200, 1800**

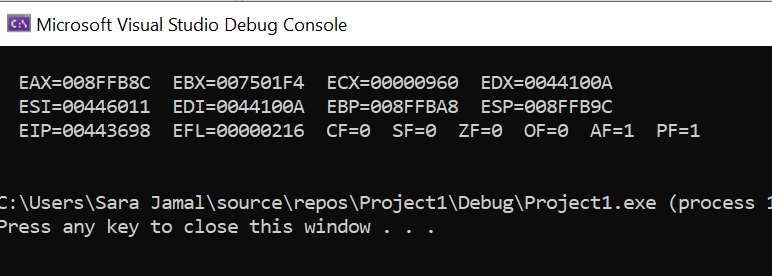
**For each array, add its 1st and last element using above 3 methods and display the result in a separate register.**

**Program:**

****

; K190207 Sara Jamal sec C

**Output:**

****

**Activity 4**

**Write down the value of each destination operand???**

**.data**

**varB BYTE 65h,33h,02h,05h**

**varW WORD 654Ah,1202h**

**varD DWORD 12344678h**

**.code**

**mov ax,WORD PTR [varB+2] 🡪 ax=0502h**

**mov bl,BYTE PTR varD 🡪 bl=78h**

**mov bl,BYTE PTR [varW+2] 🡪 bl=02h**

**mov ax,WORD PTR [varD+2] 🡪 ax=1234h**

**mov eax,DWORD PTR varW 🡪 eax=1202654Ah**