

# COAL LAB 5 - 24K0546

## Task 1

```
1 INCLUDE Irvine32.inc
2
3
4 .code
5     main PROC
6         mov AL, 7Fh
7         add AL, 1
8         call DumpRegs
9     exit
10    main ENDP
11    END main
```

Microsoft Visual Studio Debug Console

EAX=00DCF980 EBX=00E62000 ECX=003510AA EDX=003510AA  
ESI=003510AA EDI=003510AA EBP=00DCF954 ESP=00DCF948  
EIP=00353669 EFL=00000A92 CF=0 SF=1 ZF=0 OF=1 AF=1 PF=0

C:\Users\k240546\source\repos\lab02\Debug\lab02.exe (process 11624)  
To automatically close the console when debugging stops, enable Tool  
le when debugging stops.  
Press any key to close this window . . .

**ZF = 0, SF = 1, CF = 0, and OF = 1**

INCLUDE Irvine32.inc

.code

main PROC

mov AL, 7Fh

add AL, 1

call DumpRegs

exit

main ENDP

END main

```
INCLUDE Irvine32.inc

.code
    main PROC
        mov AL, 7Fh
        sub AL, 80h
        call DumpRegs
    exit
    main ENDP
END main
```

Microsoft Visual Studio Debug Console

EAX=00F3F9FF EBX=00DD6000 ECX=001410AA EDX=001410AA  
ESI=001410AA EDI=001410AA EBP=00F3F910 ESP=00F3F904  
EIP=00143669 EFL=00000A87 CF=1 SF=1 ZF=0 OF=1 AF=0 PF=1

C:\Users\k240546\source\repos\lab02\Debug\lab02.exe (process 8728)  
To automatically close the console when debugging stops, enable Tool  
le when debugging stops.  
Press any key to close this window . . .

**ZF = 0, SF = 1, CF = 1, and OF = 1**

INCLUDE Irvine32.inc

.code

main PROC

mov AL, 7Fh

sub AL, 80h

call DumpRegs

exit

main ENDP  
END main

## Task 2

```
EAX=012FFF94 EBX=01001000 ECX=00CC10AA EDX=00CC10AA
ESI=00CC6000 EDI=00CC10AA EBP=012FFF48 ESP=012FFF3C
EIP=00CC366A EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1

EAX=012FFF94 EBX=01001000 ECX=00CC10AA EDX=00CC10AA
ESI=00CC6001 EDI=00CC10AA EBP=012FFF48 ESP=012FFF3C
EIP=00CC367A EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1

EAX=012FFF94 EBX=01001000 ECX=00CC10AA EDX=00CC10AA
ESI=00CC6003 EDI=00CC10AA EBP=012FFF48 ESP=012FFF3C
EIP=00CC367E EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1

EAX=012F1234 EBX=01001000 ECX=00CC10AA EDX=00CC10AA
ESI=00CC6003 EDI=00CC10AA EBP=012FFF48 ESP=012FFF3C
EIP=00CC3689 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1

EAX=012F1234 EBX=01000001 ECX=00CC10AA EDX=00CC10AA
ESI=00CC6003 EDI=00CC10AA EBP=012FFF48 ESP=012FFF3C
EIP=00CC3692 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1

EAX=012F1234 EBX=01000002 ECX=00CC10AA EDX=00CC10AA
ESI=00CC6003 EDI=00CC10AA EBP=012FFF48 ESP=012FFF3C
EIP=00CC3698 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1

EAX=012F1234 EBX=01000004 ECX=00CC10AA EDX=00CC10AA
ESI=00CC6003 EDI=00CC10AA EBP=012FFF48 ESP=012FFF3C
EIP=00CC36A4 EFL=00000246 CF=0 SF=0 ZF=1 OF=0 AF=0 PF=1

C:\Users\user\source\repos\Lab5\Debug\Lab5.exe (process 14668) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging
Press any key to close this window . . .
```

```
INCLUDE Irvine32.inc
.data
    myByte  BYTE 12h
    myWord  WORD 1234h
    myDword DWORD 12345678h

.code
main PROC
    ;a
    MOV ESI, OFFSET myByte
    call DumpRegs
    MOV ESI, OFFSET myWord
    call DumpRegs
    MOV ESI, OFFSET myDword
    call DumpRegs

    ;b
    MOV AX, WORD PTR myDword + 2
    call DumpRegs

    ;c
    MOV BX, TYPE myByte
    call DumpRegs
    MOV BX, TYPE myWord
    call DumpRegs
    MOV BX, TYPE myDword
    call DumpRegs

    exit
main ENDP
END main
```

✓ No issues found

```
INCLUDE Irvine32.inc
```

```
.data
```

```
myByte BYTE 12h
```

```
myWord WORD 1234h
```

```
myDword DWORD 12345678h
```

```
.code
```

```
main PROC
```

```
    ;a)
```

```
    MOV ESI, OFFSET myByte
```

```
    MOV ESI, OFFSET myWord
```

```
    MOV ESI, OFFSET myWord
```

```
    ;b)
```

```
    MOV AX, WORD PTR myDword + 2
```

```
    ;c)
```

```
    MOV BX, TYPE myByte
```

```
    MOV BX, TYPE myWord
```

```
    MOV BX, TYPE myDword
```

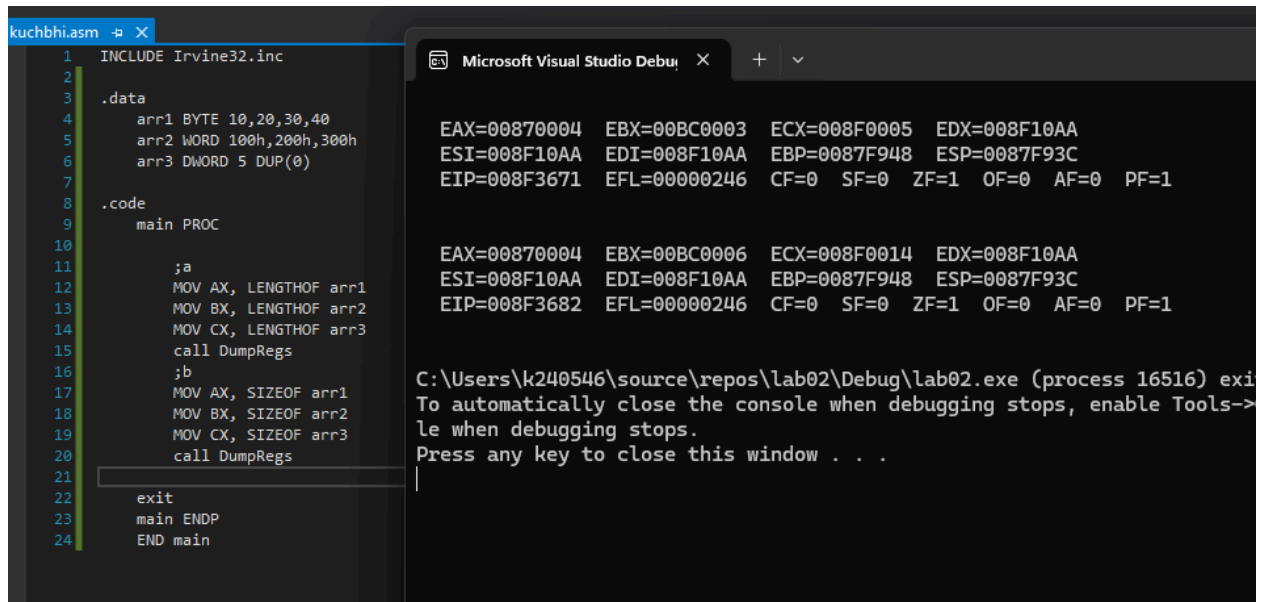
```
    call DumpRegs
```

```
exit
```

```
main ENDP
```

```
END main
```

### Task 3



The screenshot shows the Visual Studio IDE with two panes. The left pane displays the assembly file 'kuchbbhi.asm' with the following code:

```
1  INCLUDE Irvine32.inc
2
3  .data
4      arr1 BYTE 10,20,30,40
5      arr2 WORD 100h,200h,300h
6      arr3 DWORD 5 DUP(0)
7
8  .code
9      main PROC
10
11          ;a
12          MOV AX, LENGTHOF arr1
13          MOV BX, LENGTHOF arr2
14          MOV CX, LENGTHOF arr3
15          call DumpRegs
16          ;b
17          MOV AX, SIZEOF arr1
18          MOV BX, SIZEOF arr2
19          MOV CX, SIZEOF arr3
20          call DumpRegs
21
22      exit
23      main ENDP
24      END main
```

The right pane shows the 'Microsoft Visual Studio Debug' window with the following register values:

Register	Value
EAX	00870004
EBX	00BC0003
ECX	008F0005
EDX	008F10AA
ESI	008F10AA
EDI	008F10AA
EBP	0087F948
ESP	0087F93C
EIP	008F3671
EFL	00000246
CF	0
SF	0
ZF	1
OF	0
AF	0
PF	1

Below the registers, the following text is displayed:

```
C:\Users\k240546\source\repos\lab02\Debug\lab02.exe (process 16516) exit
To automatically close the console when debugging stops, enable Tools->
le when debugging stops.
Press any key to close this window . . .
```

INCLUDE Irvine32.inc

.data

```
arr1 BYTE 10,20,30,40
arr2 WORD 100h,200h,300h
arr3 DWORD 5 DUP(0)
```

.code

main PROC

```
;a
MOV AX, LENGTHOF arr1
MOV BX, LENGTHOF arr2
MOV CX, LENGTHOF arr3
call DumpRegs
;b
MOV AX, SIZEOF arr1
MOV BX, SIZEOF arr2
MOV CX, SIZEOF arr3
call DumpRegs
```

```
exit
main ENDP
END main
```

## Task 4

The screenshot shows a debugger window with two panes. The left pane displays assembly code for a program named 'kuchbhi.asm'. The code is as follows:

```
7 .code
8 main PROC
9
10 ;a
11 MOV ESI, OFFSET arrayB
12 MOV AX, [esi]
13 call DumpRegs
14
15 INC ESI
16 MOV AX, [esi]
17 call DumpRegs
18
19 INC ESI
20 MOV AX, [esi]
21 call DumpRegs
22
23 ;b
24 MOV ESI, OFFSET arrayW
25 MOV AX, [esi]
26 call DumpRegs
27
28 ADD ESI, 2
29 MOV AX, [esi]
30 call DumpRegs
31
32 ADD ESI, 2
33 MOV AX, [esi]
34 call DumpRegs
35
36 ;c DIFFERENT INCREMENT IN ESI
37
38 exit
```

The right pane shows the register values at various points in the execution. The registers are EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP, EIP, EFL, CF, SF, ZF, OF, AF, and PF. The values change as the program executes, reflecting the state of the registers at each 'DumpRegs' call.

### C. DIFFERENT INCREMENT IN ESI BECAUSE SIZE OF BOTH ARRAYS IS DIFFERENT (ArrayB is 1 byte only, while ArrayW is 2 bytes)

INCLUDE Irvine32.inc

.data

```
arrayB BYTE 11h,22h,33h
arrayW WORD 4444h,5555h,6666h
```

.code

```
main PROC

; a
MOV ESI, OFFSET arrayB
MOV AX, [esi]
call DumpRegs

INC ESI
MOV AX, [esi]
call DumpRegs

INC ESI
MOV AX, [esi]
```

```
call DumpRegs
```

```
;b
```

```
MOV ESI, OFFSET arrayW
```

```
MOV AX, [esi]
```

```
call DumpRegs
```

```
ADD ESI,2
```

```
MOV AX, [esi]
```

```
call DumpRegs
```

```
ADD ESI, 2
```

```
MOV AX, [esi]
```

```
call DumpRegs
```

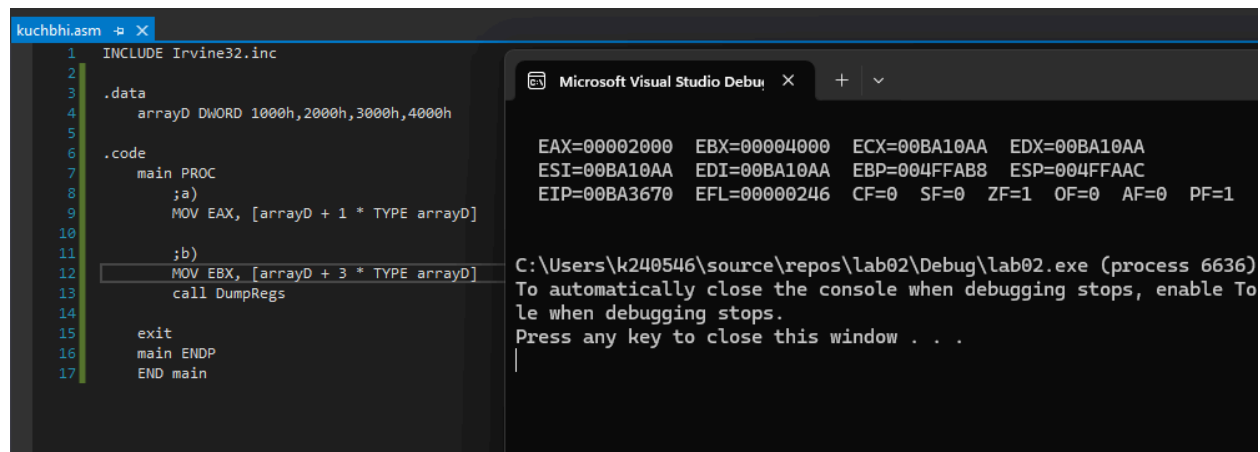
;c DIFFERENT INCREMENT IN ESI BECAUSE SIZE OF BOTH ARRAYS IS DIFFERENT (ArrayB is 1 byte only, while ArrayW is 2 bytes)

```
exit
```

```
main ENDP
```

```
END main
```

## Task 5



The screenshot shows a Visual Studio IDE with two windows. The left window, titled 'kuchbhi.asm', contains assembly code. The right window, titled 'Microsoft Visual Studio Debug Console', shows the state of registers and the execution path.

```
1  INCLUDE Irvine32.inc
2
3  .data
4      arrayD DWORD 1000h,2000h,3000h,4000h
5
6  .code
7      main PROC
8          ;a)
9          MOV EAX, [arrayD + 1 * TYPE arrayD]
10
11          ;b)
12          MOV EBX, [arrayD + 3 * TYPE arrayD]
13          call DumpRegs
14
15      exit
16      main ENDP
17      END main
```

The debug console shows the following register values:

Register	Value
EAX	00002000
EBX	00004000
ECX	00BA10AA
EDX	00BA10AA
ESI	00BA10AA
EDI	00BA10AA
EBP	004FFAB8
ESP	004FFAAC
EIP	00BA3670
EFL	00000246
CF	0
SF	0
ZF	1
OF	0
AF	0
PF	1

The console also shows the execution path: C:\Users\k240546\source\repos\lab02\Debug\lab02.exe (process 6636). Below the path, it says: 'To automatically close the console when debugging stops, enable To le when debugging stops. Press any key to close this window . . .'

**c) The type operator returns the size in bytes of one element in the array thus allows correct calculation of the byte offset for indexed addressing. It makes sure that the scaled index points precisely to the desired array element.**

```
INCLUDE Irvine32.inc
```

```
.data
```

```
arrayD DWORD 1000h,2000h,3000h,4000h
```

.code

main PROC

    ;a)

    MOV EAX, [arrayD + 1 \* TYPE arrayD]

    ;b)

    MOV EBX, [arrayD + 3 \* TYPE arrayD]

    call DumpRegs

exit

main ENDP

END main