**LAB 4**

**COAL LAB**

21k-4834

QUESTION 1

INCLUDE Irvine32.inc

.data

V1 SDWORD 8000h

.code

main PROC



mov eax,V1

add eax,1

call DumpRegs

exit

main ENDP

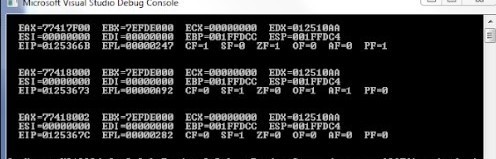
END main

QUESTION 2

INCLUDE Irvine32.inc

.code

main PROC



mov ax,7FF0h

add al,10h

add ah,1

add ax,2

call DumpRegs

exit

main ENDP

END main

QUESTION 3

INCLUDE Irvine32.inc

.data

v DWORD 8h,5h,1h,2h,6h

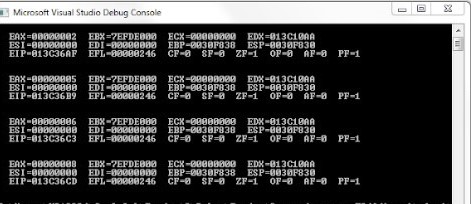
.code

main PROC

mov eax,v

xchg eax,[v+8]

mov v,eax



call DumpRegs

mov eax,v+4

xchg eax,v+12

mov v+4,eax

call DumpRegs

mov eax,v+8

xchg eax,v+12

mov v+4,eax

call DumpRegs

mov eax,v+12

xchg eax,v+16

mov v+12,eax

call DumpRegs

exit

main ENDP

END main

QUESTION 4

INCLUDE Irvine32.inc

.data

arr1 byte 10,20,30

arr2 word 150,250,350

arr3 dword 600,1200,1800

s1 DWORD 0

s2 DWORD 0

s3 DWORD 0

.code

main PROC

movzx eax,arr1

movzx ebx,arr2

add eax,ebx

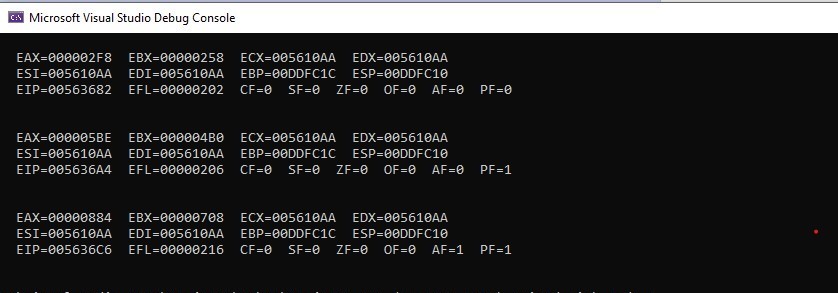
mov ebx,arr3

add eax,ebx

mov s1,eax

call DumpRegs

movzx eax,arr1+1



movzx ebx,arr2+2

add eax,ebx

mov ebx,arr3+4

add eax,ebx

mov s2,eax

call DumpRegs

movzx eax,arr1+2

movzx ebx,arr2+4

add eax,ebx

mov ebx,arr3+8

add eax,ebx

mov s3,eax

call DumpRegs

exit

main ENDP

END main

QUESTION 5

INCLUDE Irvine32.inc

.data

arr1 BYTE 10h, 20h, 30h, 40h

arr2 BYTE 4 DUP (?)

.code

main PROC

mov eax,0

mov al,arr1+3

mov arr2,al

mov al,arr1+2

mov arr2+1,al

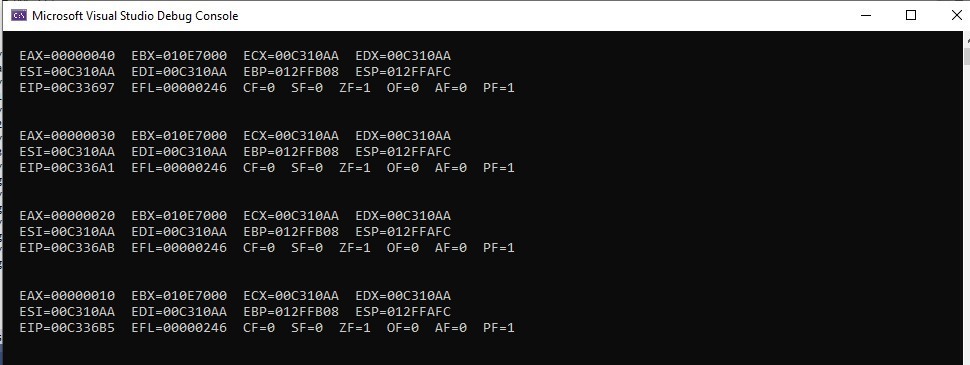
mov al,arr1+1

mov arr2+2,al

mov al,arr1

mov arr2+3,al

mov al,arr2



call DumpRegs

mov al,arr2+1

call DumpRegs

mov al,arr2+2

call DumpRegs

mov al,arr2+3

call DumpRegs

exit

main ENDP

END main

QUESTION 6

INCLUDE Irvine32.inc

.data

arr DWORD 5000h,500h,150h,300h,1000h

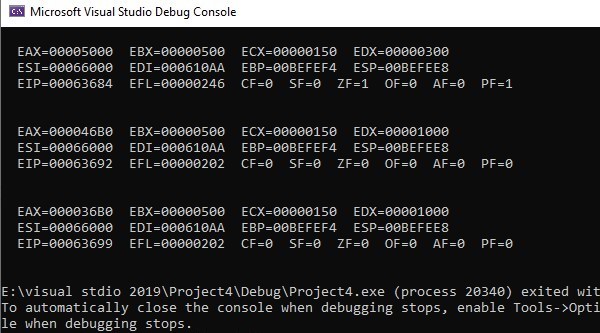
.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0



mov edx, 0

mov esi, OFFSET arr

mov eax, [esi]

mov ebx, [esi+4]

mov ecx, [esi+8]

mov edx, [esi+12]

call DumpRegs

sub eax,ebx

sub eax,ecx

sub eax,edx

mov edx, [esi+16]

call DumpRegs

sub eax,edx

call DumpRegs

exit

main ENDP

END main

QUESTION 7

INCLUDE Irvine32.inc

.data

arr1 BYTE 60, 70, 80

arr2 WORD 150, 250, 350

arrD DWORD 600, 1200, 1800

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx,0

mov esi, 0

movzx eax,arr1[esi \* TYPE arr1]

mov esi,2

movzx ebx,arr1[esi \* TYPE arr1]

add eax,ebx

mov esi, 0

movzx eax,arr2[esi \* TYPE arr2]

mov esi,2

movzx ebx,arr2[esi \* TYPE arr2]

add ebx,eax

mov esi, 0

mov eax,arr3[esi \* TYPE arr3]

mov esi,2

mov ecx,arr3[esi \* TYPE arr3]

add ecx,eax

call DumpRegs

exit

main ENDP

END main

