# COAL LAB 7

Q.1)

Include Irvine32.inc

.data

var dword 5

x dword 0

.code

main proc

call dumpregs

mov eax,var

cmp eax,ecx

jl Second\_condition

jmp else\_

Second\_condition:

cmp ecx,edx

jl else\_

mov ebx,0

mov x,ebx

jmp exitt

else\_:

mov ebx,x

add ebx,1

mov x,ebx

exitt:

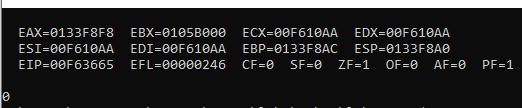
mov eax,x

call writedec

exit

main ENDP

END main



Q.2)

Include Irvine32.inc

.data

intArr SWORD 0, 0, 0, 0, 1, 20, 35, -12, 66, 4, 0

.code

main proc

mov eax,0

mov esi,offset intArr

continue:

mov bx,[esi]

cmp ax,bx

call dumpregs

jnz print

add esi,2

jmp continue

print:

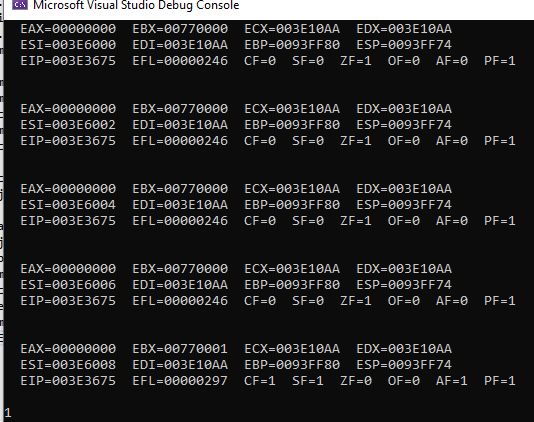
mov ax,[esi]

call writedec

exit

main ENDP

END main



Q.3)

Include Irvine32.inc

.data

msg1 byte "Enter 4 Integers: ",0

msg2 byte "Not Equal",0

msg3 byte "Equal",0

intArr DWORD 4 DUP(?)

.code

main proc

mov edx,offset msg1

call writestring

mov eax,0

mov esi,offset intArr

mov ecx,4

L1:

call Readint

mov [esi],eax

add esi,4

loop L1

mov esi,offset intArr

mov ecx,3

continue:

mov eax,[esi]

mov ebx,[esi+4]

cmp eax,ebx

call dumpregs

jnz print

add esi,4

loop continue

mov edx,offset msg3

call writestring

jmp exitt

print:

mov edx,offset msg2

call writestring

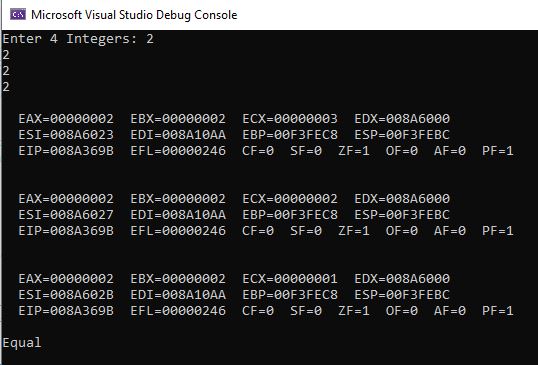
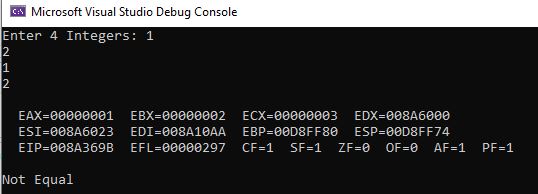
exitt:

call crlf

exit

main ENDP

END main



Q.4)

Include Irvine32.inc

.data

arr DWORD 10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20

msg byte "Enter number for search:",0

msg1 byte "Found",0

msg2 byte "Not found",0

.code

main proc

mov edx,offset msg

call writestring

call readint

mov esi,offset arr

mov ecx,lengthof arr

continue:

mov ebx,[esi]

cmp eax,ebx

jz print

add esi,4

loop continue

mov edx,offset msg2

call writestring

jmp exitt

print:

mov edx,offset msg1

call writestring

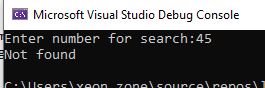
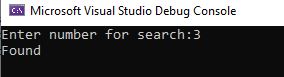
exitt:

call crlf

exit

main ENDP

END main



Q.5)

Include Irvine32.inc

.data

arr DWORD 10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20

Swap\_Count DWORD 0

msg byte " swaps",0

.code

main proc

mov edx,0

mov eax,lengthof arr

sub eax,1

mov esi,offset arr

mov ecx,eax

continue:

mov eax,[esi]

mov ebx,[esi+4]

cmp eax,ebx

jle no\_swap

call writeint

add edx,1

mov [esi+4],eax

mov [esi],ebx

no\_swap:

add esi,4

loop continue

call crlf

mov Swap\_Count,edx

mov eax,Swap\_count

call writedec

mov edx,offset msg

call writestring

call crlf

exit

main ENDP

END main

