**COAL LAB 7**

**Question 1**

Include Irvine32.inc

.data

Var DWORD 5

X DWORD ?

.code

Main proc

MOV ecx, 4

MOV edx, 7

Cmp var, ecx

JAE elsee

Cmp ecx, edx

Jb elsee

MOV x, 0

Jmp endd

elsee:

MOV x, 1

endd:

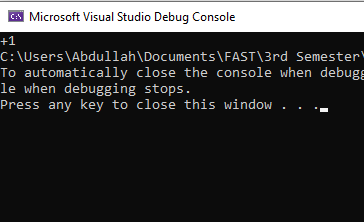
Mov eax, x

call writeint

exit

main endp

end main

****

**Question 2**

Include Irvine32.inc

.data

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

str1 BYTE "The first non zero value in the array is: ", 0

.code

Main PROC

MOV esi, OFFSET intARR

MOV eax, 0

MOV ebx, 0

L1:

MOVSX ebx, SWORD ptr [esi]

cmp eax, ebx

JNE endd

ADD esi, type intArr

jmp L1

endd:

mov edx, OFFSET str1

MOVSX eax, SWORD ptr [esi]

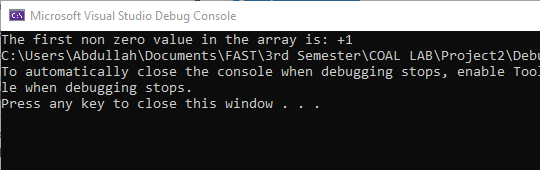
call WriteString

call Writeint

exit

main endp

end main

****

**Question 3**

Include Irvine32.inc

.data

str1 Byte "Enter number #" , 0

str2 Byte "The numbers are not equal.", 0

str3 Byte "The numbers are equal.", 0

arr1 DWORD 4 dup(?)

.code

main PROC

MOV eax, 1

MOV ecx, 4

MOV esi, OFFSET arr1

MOV edx, OFFSET str1

input:

call writestring

call writedec

push eax

MOV al, ' '

call writechar

call readint

MOV [esi], eax

ADD esi, 4

pop eax

inc eax

call crlf

loop input

MOV eax, arr1

MOV ebx, [arr1 + 4]

MOV ecx, [arr1 + 8]

MOV edx, [arr1 + 12]

cmp eax, ebx

JE cond2

Jmp endd

cond2:

cmp eax, ecx

JE cond3

jmp endd

cond3:

cmp eax, edx

JE output

Jmp endd

output:

mov edx, OFFSet str3

call writestring

exit

endd:

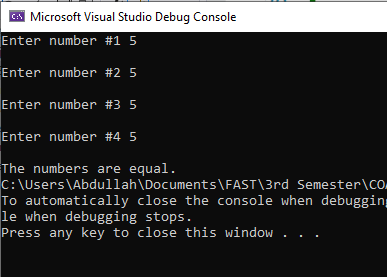
MOV edx, OFFSET str2

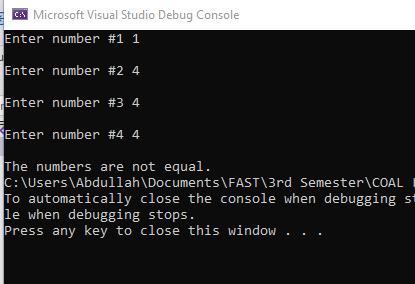
call writestring

exit

main endp

end main

****

****

**Question 4**

Include Irvine32.inc

.data

Str1 BYTE "Enter a number to search: ", 0

Str2 BYTE "The number occurs in the Array.", 0

Str3 BYTE "The number does not occur in the Array. ", 0

Arr WORD 10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20

.code

Main PROC

MOV eax, 0

MOV edx, OFFSET str1

Call WriteString

Call Readdec

call crlf

MOV esi, OFFSET Arr

MOV ecx, lengthof Arr

L1:

Cmp ax, [esi]

JE endd

ADD esi, type Arr

loop l1

Mov edx, offset str3

call writestring

exit

endd:

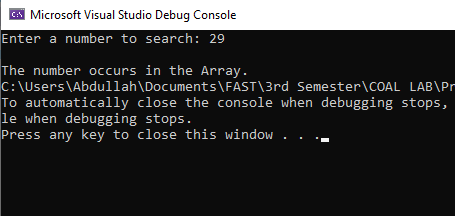
Mov edx, offset str2

call writestring

exit

main endp

end main

****

**Question 5**

Include Irvine32.inc

.data

swap\_count DWORD 0

arr1 SDWORD 6, 1, -74, 35, -56

.code

main PROC

MOV ecx, 4

MOV esi, OFFSET arr1

MOV eax, 0

l1:

MOV eax, [esi]

cmp eax, [esi + 4]

JLE next

call swap

inc swap\_count

next:

ADD esi, type arr1

loop l1

endd:

MOV eax, swap\_count

call writedec

exit

swap:

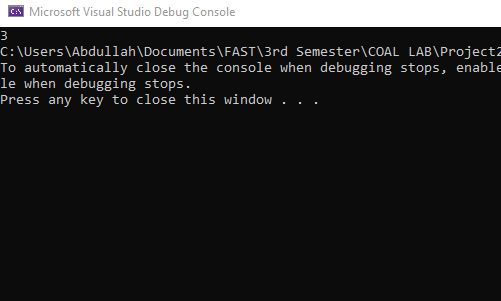
XCHG eax, [esi + 4]

MOV [esi], eax

ret

main endp

end main

****