23I-0544

MUHAMMAD HAMMAD

BCS-3D

COAL LAB 08

TASK 01:  
;Write a program that takes four input integers from the user.

;Then compare and display a message whether these integers

;are equal or not.

INCLUDE Irvine32.inc

.data

prompt1 BYTE "Enter 4 integers: " , 0

prompt2 BYTE "The integers are equal: ", 0

prompt3 BYTE "The integers are not equal: " ,0

one DWORD ?

two DWORD ?

three DWORD ?

four DWORD ?

.code

main PROC

mov edx, OFFSET prompt1

call writestring

call crlf

mov edx, 0

call readint

mov one, edx

mov edx, 0

call readint

mov two, edx

mov edx, 0

call readint

mov three, edx

mov edx, 0

call readint

mov four, edx

mov edx, one

cmp edx, two

JNE ex

mov edx, two

cmp edx, three

JNE ex

mov edx, three

cmp edx, four

JNE ex

mov edx, OFFSET prompt2

call writestring

jmp done

ex:

mov edx, OFFSET prompt3

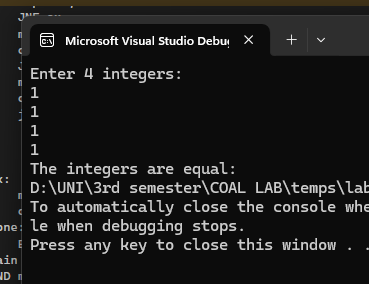
call writestring

done:

EXIT

main ENDP

END main

  
  
TASK 02:  
INCLUDE Irvine32.inc

.data

prompt1 BYTE "First non-zero value found: ", 0

intArr SWORD 0, 0, 0, 150, 120, 35, -12, 66, 4, 0

.code

main PROC

mov esi, OFFSET intArr

mov cx, LENGTHOF intArr

mov eax, 0

L:

mov ax, [esi]

cmp ax, 0

JNZ found

add esi, 2

loop L

jmp done

found:

mov edx, OFFSET prompt1

call writestring

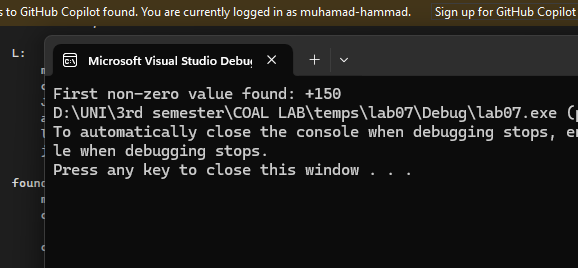
call writeint

done:

EXIT

main ENDP

END main

  
  
TASK 003:  
;Task#3 Implement the following given code in Assembly and Consider

;var = 5 , edx = var+1 and counter value from array initialized in task#2.

;if ( var < ecx ) AND (ecx >= edx) then

;x = 0

;else

;x = 1

INCLUDE Irvine32.inc

.data

prompt1 BYTE "value of X is: ", 0

intArr SWORD 0, 0, 0, 150, 120, 35, -12, 66, 4, 0

var DWORD 5

X DWORD ?

.code

main PROC

mov cx, LENGTHOF intArr

L:

mov edx, var

inc edx

cmp var, ecx

JGE notzero

cmp ecx, edx

JNGE notzero

mov edx, OFFSET prompt1

mov eax, 0

call writestring

call writedec

call crlf

jmp ex

notzero:

mov eax, 1

mov edx, OFFSET prompt1

call writestring

call writedec

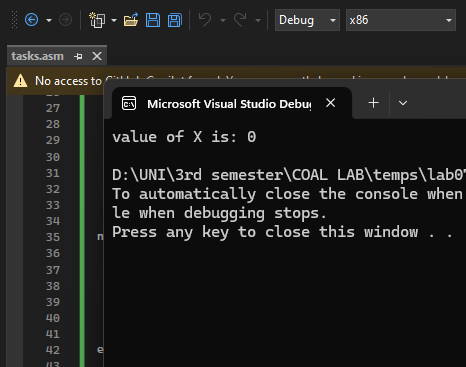
call crlf

ex:

EXIT

main ENDP

END main

  
  
TASK 04:  
;Task#4 Implement the following given code in Assembly and consider var = 0.

;while ( var <= 10)

;if (var < 5)

;Print “Hello”

;else

;Print “World”

;var = var + 1

;end while

INCLUDE Irvine32.inc

.data

prompt1 BYTE "Hello" , 0

prompt2 BYTE "World" , 0

var DWORD 0

.code

main PROC

mov ecx, 0

again:

cmp var, 10

JE ex

cmp var, 5

JGE hello

mov edx, OFFSET prompt2

call writestring

call crlf

inc var

jmp again

hello:

mov edx, OFFSET prompt1

call writestring

call crlf

inc var

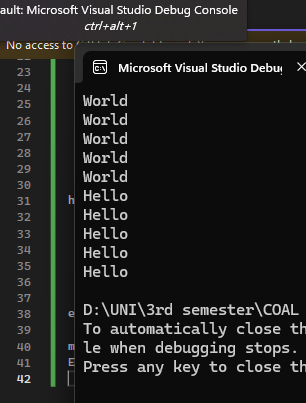
jmp again

ex:

EXIT

main ENDP

END main

  
  
TASK 05:  
;Task#5

;Write a program for sequential search. Take an input from the user and find if it occurs in

;the following array:

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

INCLUDE Irvine32.inc

.data

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

arr2 WORD 11 DUP(?)

prompt1 BYTE "Enter the number to search from this array (10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20): ", 0

prompt2 BYTE "Number not found in the sequence!", 0

prompt3 BYTE "Number found in the sequence", 0

.code

main PROC

mov ecx, 11

mov edx, OFFSET prompt1

call writestring

call readint

call crlf

mov esi, OFFSET arr

L:

movzx ebx, WORD PTR [esi]

cmp eax, ebx

JE found

ADD esi, 2

loop L

mov edx, OFFSET prompt2

call writestring

jmp ex

found:

mov edx, OFFSET prompt3

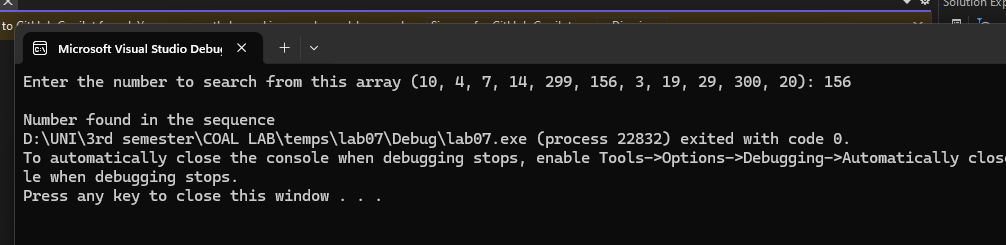
call writestring

ex:

EXIT

main ENDP

END main

  
  
  
TASK 07:

;Write a program to print weekday based on given number

INCLUDE Irvine32.inc

.data

Monday BYTE "Monday" , 0

Tuseday BYTE "Tuseday" , 0

Wednesday BYTE "Wednesday" , 0

Thursday BYTE "Thursday" , 0

Friday BYTE "Friday" , 0

Saturday BYTE "Saturday" , 0

Sunday BYTE "Sunday", 0

prompt1 BYTE "Enter the number of day (1-7): ", 0

.code

main PROC

mov edx, OFFSET prompt1

call writestring

call readint

cmp eax, 1

JE monday1

cmp eax, 2

JE tuseday2

cmp eax, 3

JE wednesday3

cmp eax, 4

JE thursday4

cmp eax, 5

JE friday5

cmp eax, 6

JE saturday6

cmp eax, 7

JE sunday7

monday1:

mov edx, OFFSET Monday

call writestring

jmp ex

tuseday2:

mov edx, OFFSET Tuseday

call writestring

jmp ex

wednesday3:

mov edx, OFFSET Wednesday

call writestring

jmp ex

thursday4:

mov edx, OFFSET Thursday

call writestring

jmp ex

friday5:

mov edx, OFFSET Friday

call writestring

jmp ex

saturday6:

mov edx, OFFSET Saturday

call writestring

jmp ex

sunday7:

mov edx, OFFSET Sunday

call writestring

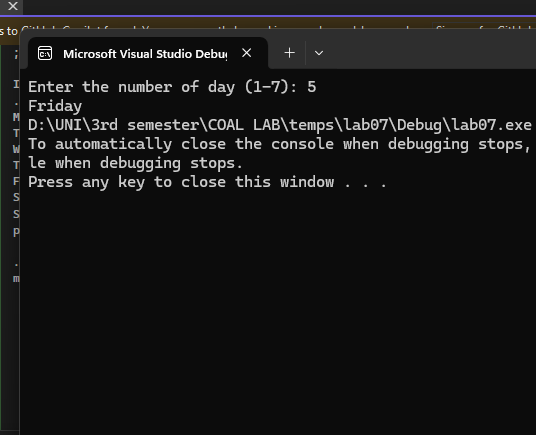
jmp ex

ex:

EXIT

main ENDP

END main



TASK 08:  
;Write a program to check whether a character is alphabet or not.

INCLUDE Irvine32.inc

.data

prompt1 BYTE "Enter the character: ", 0

prompt2 BYTE " is alphabet", 0

prompt3 BYTE " is not alphabet", 0

.code

main PROC

mov edx, OFFSET prompt1

call writestring

call readchar

call crlf

call writechar

;checking for upper case

cmp al, 'A'

JL lower

cmp al, 'Z'

JG lower

jmp alphabet

lower:;checking for lower case

cmp al, 'a'

JL notalphabet

cmp al, 'z'

JG notalphabet

jmp alphabet

;checking for upper case

cmp al, 'A'

JL notalphabet

cmp al, 'Z'

JG notalphabet

jmp alphabet

alphabet:

mov edx, OFFSET prompt2

call writestring

jmp ex

notalphabet:

mov edx, OFFSET prompt3

call writestring

jmp ex

ex:

EXIT

main ENDP

END main

