**LAB# 07**

**Task 1 A:**

Include Irvine32.inc

.data

var DWORD 5

var1 DWORD 4

var2 DWORD 3

x DWORD ?

.code

main proc

mov eax,var

mov ecx,var1

mov edx,var2

CMP eax,ecx

JNB elseStatement

CMP eax,edx

JNAE elseStatement

JMP trueStatement

elseStatement:

mov x,1

trueStatement:

mov x,0

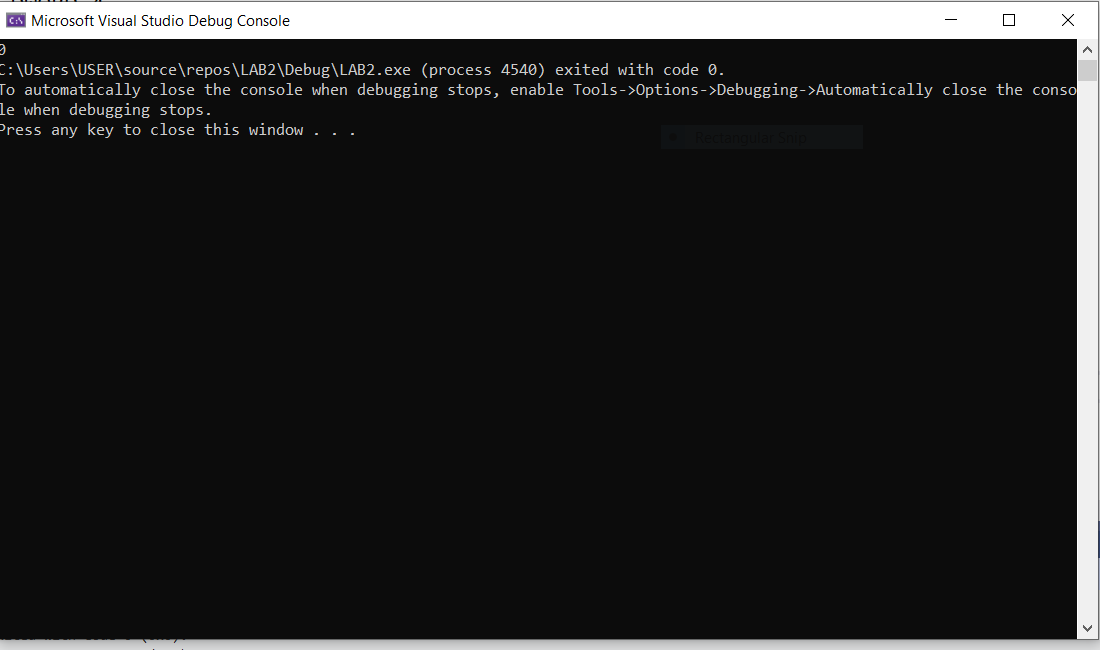
mov eax,x

call WriteDec

exit

main ENDP

END main



**B:**

Include Irvine32.inc

.data

var DWORD 0 ; Define var as a 16-bit integer variable with initial value 0

msg\_hello db "Hello$"

msg\_world db "World$"

.code

main proc

start:

mov ax, @data

mov ds, ax

mov cx, 11

mov bx, 0

while\_loop:

cmp bx, cx

jg

mov ax, [var]

mov dx, 0

div byte ptr 2

cmp dx, 0

jne

mov ah, 9

lea dx, msg\_hello

int 21h

jmp next\_iter

print\_world:

mov ah, 9

lea dx, msg\_world

int 21h

next\_iter:

add [var], 1

add bx, 1

jmp while\_loop

exit

main endp

end main

**Task 3:**

Include Irvine32.inc

.data

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

prompt BYTE "Non-zero element found: ",0

prompt2 BYTE "Non-zero Element not found",0

required SWORD 0

.code

main proc

mov eax,0

mov esi,OFFSET intArr

mov ecx, LENGTHOF intArr

traverse:

mov ax,[esi]

CMP ax,required

JNZ found

add esi, TYPE intArr

loop traverse

jmp notFound

found:

mov edx, OFFSET prompt

call WriteString

mov ax,[esi]

call WriteDec

exit

notFound:

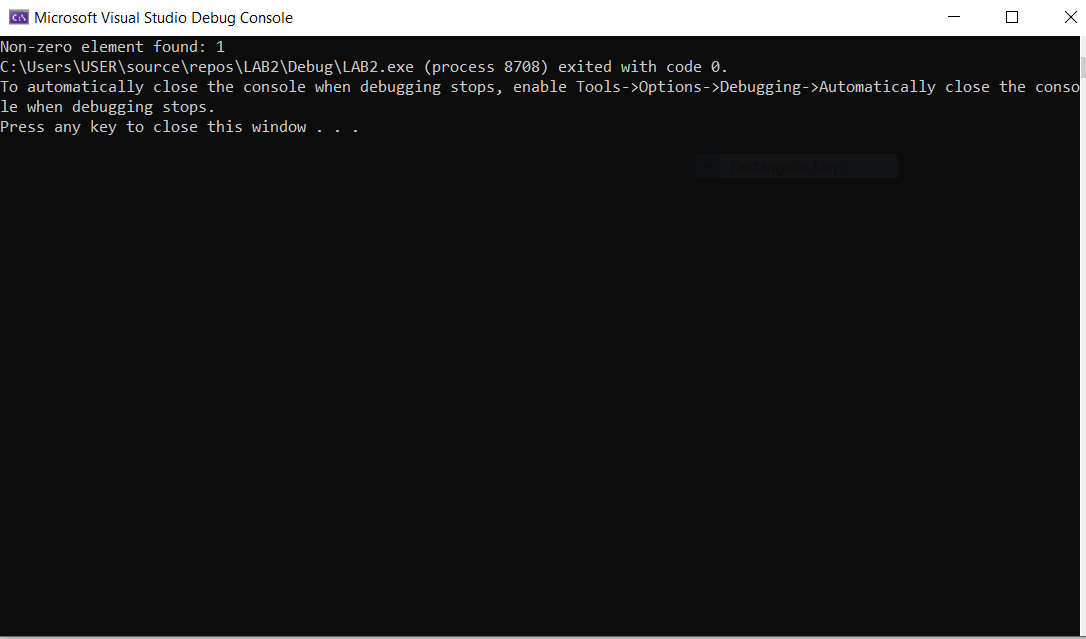
mov edx,OFFSET prompt2

call WriteString

exit

main ENDP

END main



**Task 4:**

Include Irvine32.inc

.data

Arr DWORD 4 DUP(?)

prompt BYTE "Enter Element ",0

colon BYTE " : ",0

right BYTE "The entered elements are equal",0

wrong BYTE "The entered elements are unequal",0

.code

main proc

mov eax,0

mov ebx,1

mov esi,OFFSET Arr

mov ecx,LENGTHOF Arr

getInput:

mov edx,OFFSET prompt

call WriteString

mov eax,ebx

call WriteDec

mov edx, OFFSET colon

call WriteString

call ReadInt

mov [esi],eax

add esi, TYPE Arr

inc ebx

loop getInput

mov ecx,LENGTHOF Arr-1

mov esi,OFFSET Arr

mov eax,0

checkEquality:

mov eax,[esi]

CMP eax,[esi + TYPE Arr]

JNE unEqual

add esi,TYPE Arr

loop checkEquality

Equal:

mov edx, OFFSET right

call WriteString

exit

unEqual:

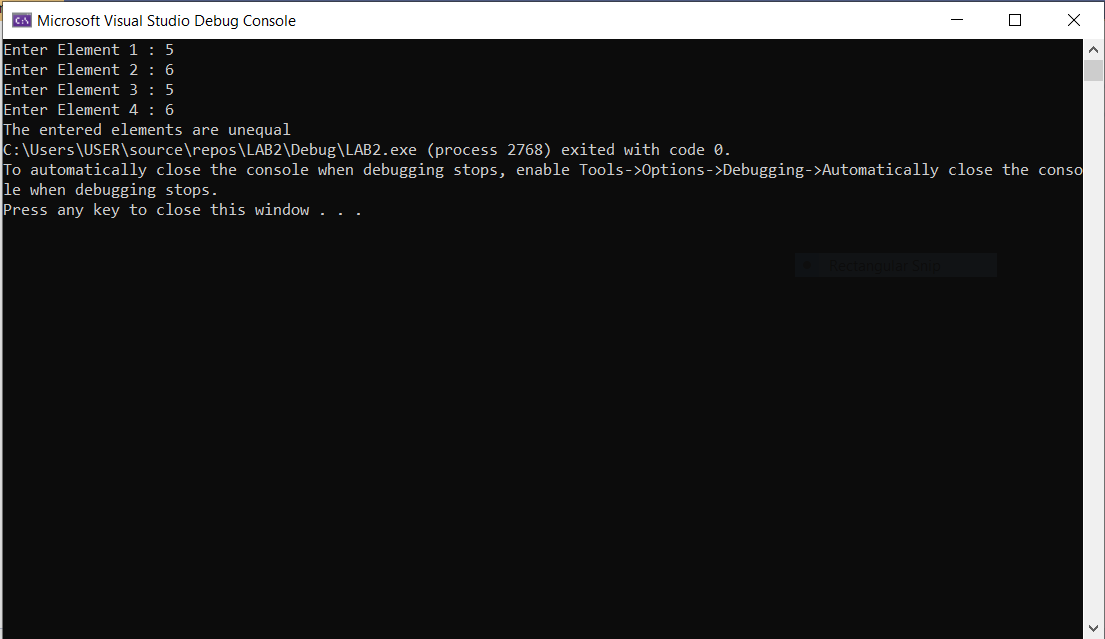
mov edx,OFFSET wrong

call WriteString

exit

main ENDP

END main



**Task 5:**

Include Irvine32.inc

.data

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

question BYTE "Enter the number: ",0

prompt BYTE "Required Element found at position: ",0

prompt1 BYTE "Required Element does not exist",0

position WORD 1

required WORD ?

.code

main proc

mov eax,0

mov esi,OFFSET arr

mov edx,OFFSET question

call WriteString

call ReadInt

mov required,ax

mov ecx, LENGTHOF arr

searchElement:

mov ax,[esi]

CMP ax,required

JE elementFound

inc position

add esi, TYPE arr

loop searchElement

notFound:

mov edx,OFFSET prompt1

call WriteString

exit

elementFound:

mov edx,OFFSET prompt

call WriteString

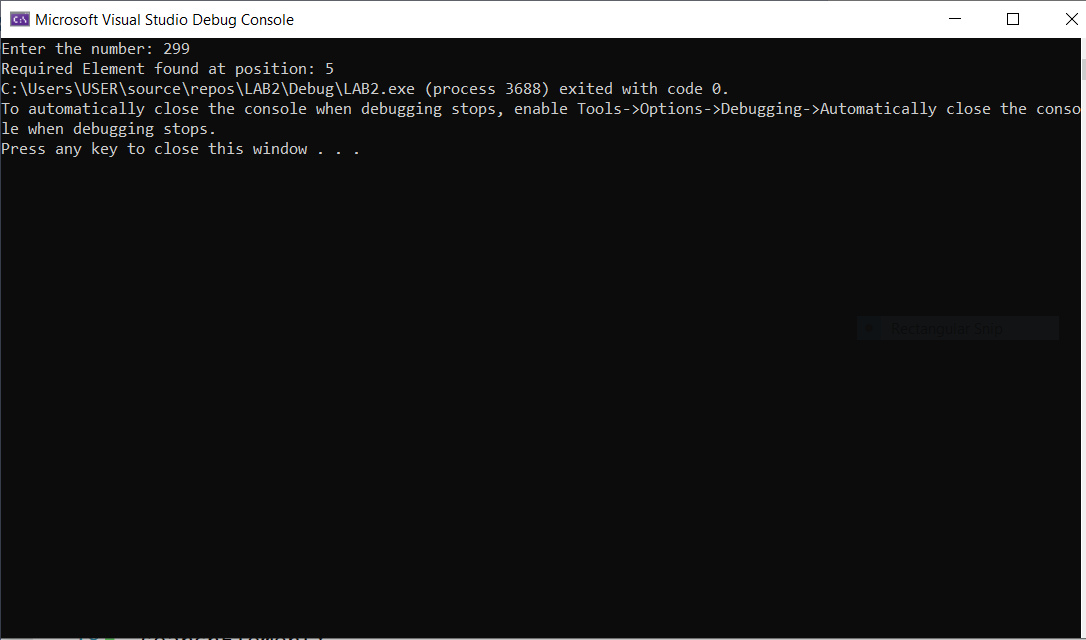
mov ax,position

call WriteDec

exit

main ENDP

END main



**Task 6:**

Include Irvine32.inc

.data

arr DWORD 10,9,8,7,6,5,4,3,2,1

Swap\_count DWORD 0

temporary DWORD ?

prompt BYTE "Number of swaps: ",0

prompt1 BYTE "Last value: ",0

.code

main proc

mov eax,0

mov esi,OFFSET arr

mov ecx,LENGTHOF arr

dec ecx

findSwaps:

mov eax,[esi]

CMP eax,[esi + 4]

JA swap

continue:

add esi,4

loop findSwaps

display:

mov edx,OFFSET prompt

call WriteString

mov eax,Swap\_count

call WriteDec

call crlf

mov edx,OFFSET prompt1

call WriteString

mov eax,[arr+36]

call WriteDec

exit

swap:

inc Swap\_count

mov temporary,eax

mov ebx,[esi+4]

mov [esi],ebx

mov [esi+4],eax

jmp continue

exit

main ENDP

END main

