Task 1

INCLUDE Irvine32.inc

.data

input\_prompt BYTE "Enter 4 Numbers ", 0

user\_input DWORD 4 DUP(?)

output\_prompt BYTE "Not Equal", 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 4

mov edx, OFFSET input\_prompt

call WriteString

call Crlf

mov edx, OFFSET user\_input

read\_loop:

call ReadInt

mov [edx], eax

add edx, 4

loop read\_loop

mov ecx, LENGTHOF user\_input - 1

mov edx, OFFSET user\_input

compare\_loop:

mov eax, [edx]

cmp eax, [edx + 4]

je found\_equal

add edx, 4

loop compare\_loop

found\_equal:

mov edx, OFFSET output\_prompt

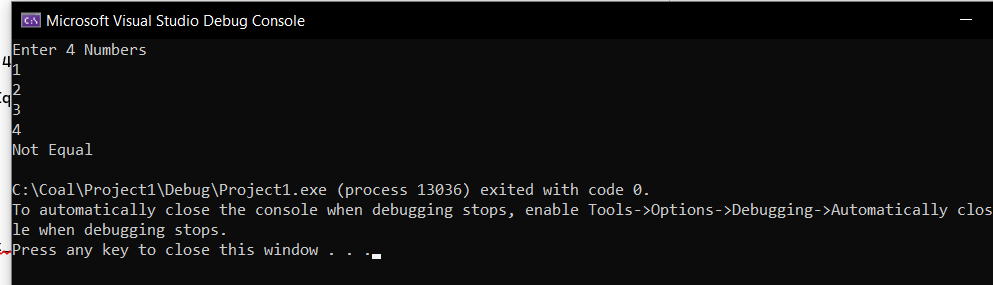
call WriteString

call Crlf

exit

main ENDP

END main



Task 2

INCLUDE Irvine32.inc

.data

intarr SWORD 0, 0, 0, 1500, 120, 35, -12, 66, 4, 0

msg BYTE "No Number", 0

msg2 BYTE "1st nonzero is ", 0

.code

main PROC

mov esi,0

mov ebx, 0

mov eax, 0

mov ecx, 10

Li:

mov ax, [intarr+esi]

cmp eax, ebx

jne L2

add esi, 2

loop Li

mov edx, OFFSET msg

call WriteString

call CRLF

jmp L3

L2:

mov edx, OFFSET msg2

call WriteString

call Writedec

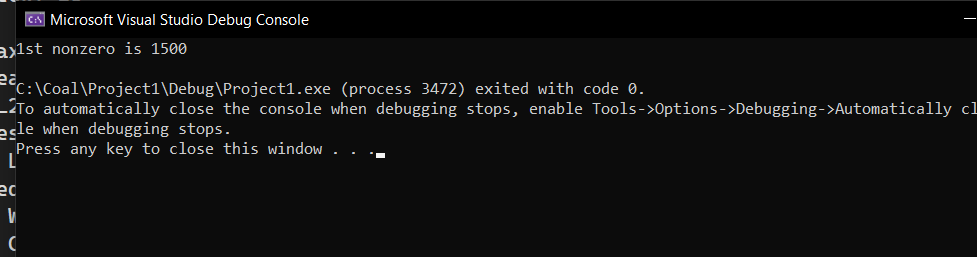
call CRLF

L3:

exit

main ENDP

end Main



Task 3

INCLUDE Irvine32.inc

.data

result\_value DWORD ?

iteration\_count = 5

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

message\_true BYTE "True ", 0

message\_false BYTE " False ", 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov eax, iteration\_count

add eax, 1

mov edx, eax

mov ecx, LENGTHOF myIntArray

mov eax, iteration\_count

cmp eax, ecx

ja second\_cond\_jump

jmp else\_block\_jump

first\_cond\_jump:

cmp ecx, edx

jge cond\_satisfied\_jump

jmp else\_block\_jump

second\_cond\_jump: ; 2nd condition

cmp ecx, edx

jge cond\_satisfied\_jump

jmp else\_block\_jump

cond\_satisfied\_jump:

mov ebx, 0

mov result\_value, ebx

cmp result\_value, 0

jnz else\_block\_jump

cond\_true\_jump:

mov edx, OFFSET message\_true

call WriteString

call Crlf

jmp quit\_jump

else\_block\_jump:

mov ebx, 1

mov result\_value, ebx

mov edx, OFFSET message\_false

call WriteString

call Crlf

jmp quit\_jump

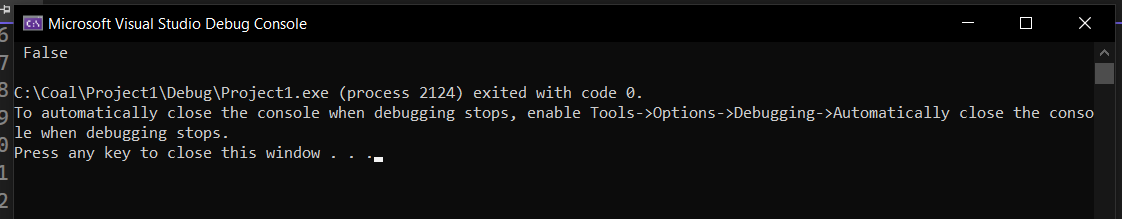
quit\_jump:

exit

main ENDP

END main

Task 4



Task 4

INCLUDE Irvine32.inc

.data

iteration\_count byte ?

txt1 byte "Hello World!",0

txt2 byte "condition is false!"

.code

main PROC

mov iteration\_count,0

continue\_jump:

CMP iteration\_count,5

JGE continue\_label

mov edx, offset txt2

call WriteString

call Crlf

jmp increment\_jump

continue\_label:

mov edx, offset txt1

call WriteString

call Crlf

increment\_jump:

inc iteration\_count

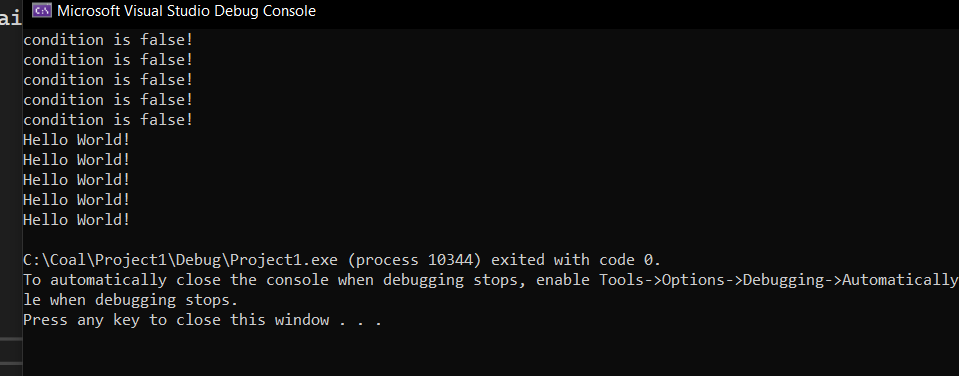
CMP iteration\_count,10

JL continue\_jump

exit

main ENDP

END main



Task 5

include irvine32.inc

.data

msg BYTE "enter int ", 0

mon BYTE "Monday", 0

tue BYTE "Tuesday", 0

wed BYTE "Wednesdsay", 0

thu BYTE "Thursday", 0

fri BYTE "Friday", 0

sat BYTE "Saturday", 0

sun BYTE "Sunday", 0

error1 BYTE "Invalid Day", 0

.code

main PROC

mov edx, OFFSET msg

call WriteString

call ReadInt

mov ebx, 1

cmp eax, ebx

jb NF

ja CheckTue

mov edx, OFFSET mon

jmp L1

CheckTue:

inc ebx

cmp eax, ebx

jb NF

ja CheckWed

mov edx, OFFSET tue

jmp L1

CheckWed:

inc ebx

cmp eax, ebx

jb NF

ja CheckThurs

mov edx, OFFSET wed

jmp L1

CheckThurs:

inc ebx

cmp eax, ebx

jb NF

ja CheckFri

mov edx, OFFSET thu

jmp L1

CheckFri:

inc ebx

cmp eax, ebx

jb NF

ja CheckSat

mov edx, OFFSET fri

jmp L1

CheckSat:

inc ebx

cmp eax, ebx

jb NF

ja CheckSun

mov edx, OFFSET sat

jmp L1

CheckSun:

inc ebx

cmp eax, ebx

jne NF

mov edx, OFFSET sun

jmp L1

NF:

mov edx, OFFSET error1

L1:

call WriteString

call CRLF

exit

main ENDP

END main

