COAL LAB TASKS

MUHAMMAD HAMMAD

23I-0544

LAB TASKS

TASK01  
INCLUDE Irvine32.inc

.data

.code

main PROC

mov eax, 0

mov ebx, 1

mov ecx, 7

call writeint

call crlf

inc eax

call writeint

call crlf

mov eax, ebx

call writeint

mov ebx, eax

call crlf

mov edx, eax

L1:

mov eax, ebx

add eax, edx

call writeint

call crlf

mov ebx, edx

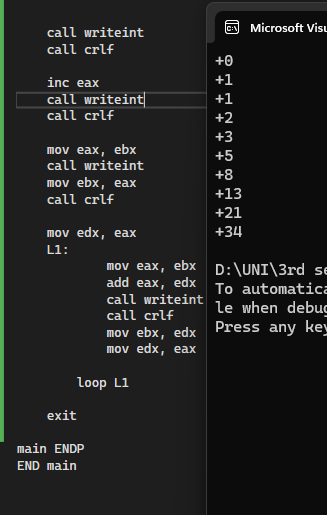
mov edx, eax

loop L1

exit

main ENDP

END main



TASK 02:

INCLUDE Irvine32.inc

.data

.code

main PROC

mov eax, 1

mov ecx, 4

mov ebx, 4

L1:

mov edx, ecx

mov ecx, ebx

L2:

call writedec

loop L2

call crlf

dec ebx

mov ecx, edx

loop L1

call crlf

mov eax, 1

mov ecx, 4

mov ebx, 1

H1:

mov edx, ecx

mov ecx, ebx

H2:

call writedec

loop H2

call crlf

inc ebx

mov ecx, edx

loop H1

call crlf

mov ecx, 4

mov ebx, 4

A1:

mov eax, 1

mov edx, ecx

mov ecx, ebx

A2:

call writedec

inc eax

loop A2

call crlf

dec ebx

mov ecx, edx

loop A1

call crlf

mov ecx, 4

mov ebx, 4

B1:

mov eax, 4

mov edx, ecx

mov ecx, ebx

B2:

call writedec

dec eax

loop B2

call crlf

dec ebx

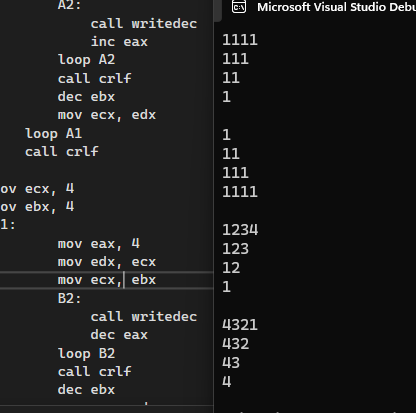
mov ecx, edx

loop B1

exit

main ENDP

END main



TASK 03:

INCLUDE Irvine32.inc

.data

nameArray BYTE 5 DUP(20 DUP(?))

idArray DWORD 5 DUP(?)

yobArray DWORD 5 DUP(?)

salaryArray DWORD 5 DUP(?)

totalSalary DWORD 0

promptName BYTE "Enter name:", 0

promptID BYTE "Enter Employee ID:", 0

promptYOB BYTE "Enter Year of Birth:", 0

promptSalary BYTE "Enter Monthly Salary:", 0

newline BYTE 0Dh, 0Ah, 0

.code

main PROC

mov ebx, 5

mov ecx, 5

mov esi, offset nameArray

label1:

mov edx, offset promptName

call WriteString

mov edx, esi

mov ecx, 20

call ReadString

add esi, 20

mov edx, offset promptID

call WriteString

call ReadInt

mov idArray[ebx \* 4 - 4], eax

mov edx, offset promptYOB

call WriteString

call ReadInt

mov yobArray[ebx \* 4 - 4], eax

mov edx, offset promptSalary

call WriteString

call ReadInt

mov salaryArray[ebx \* 4 - 4], eax

dec ebx

jnz label1

mov ebx, 5

mov esi, offset nameArray

label2:

mov edx, esi

call WriteString

call Crlf

mov edx, offset newline

call WriteString

; Display Employee ID

mov eax, idArray[ebx \* 4 - 4]

call WriteInt

call Crlf

; Display Year of Birth

mov eax, yobArray[ebx \* 4 - 4]

call WriteInt

call Crlf

; Calculate and display Annual Salary

mov eax, salaryArray[ebx \* 4 - 4]

mov edx, 12

mul edx

add totalSalary, eax

call WriteInt

call Crlf

add esi, 20

dec ebx

jnz label2

mov edx, offset newline

call WriteString

mov eax, totalSalary

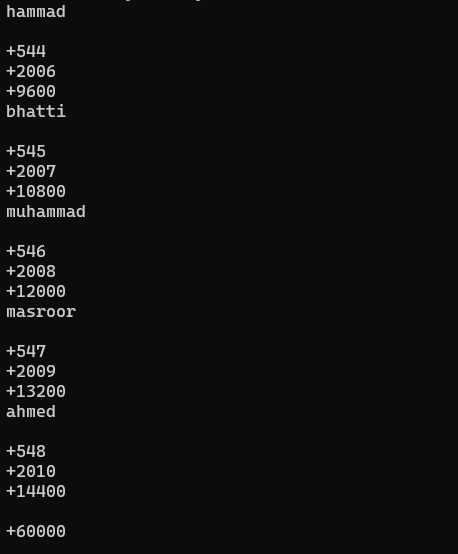
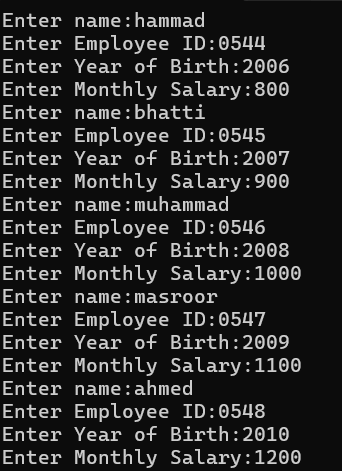
call WriteInt

call Crlf

exit

main ENDP

END main



TASK 04:  
include irvine32.inc

.data

source BYTE "Coppied Successfully" , 0

target BYTE 20 DUP(?)

.Code

MAIN PROC

mov esi, OFFSET source

mov edi, OFFSET target

copy:

mov al, [esi]

mov [edi], al

cmp al, 0

je Copied

inc esi

inc edi

jmp copy

Copied:

mov edx, OFFSET target

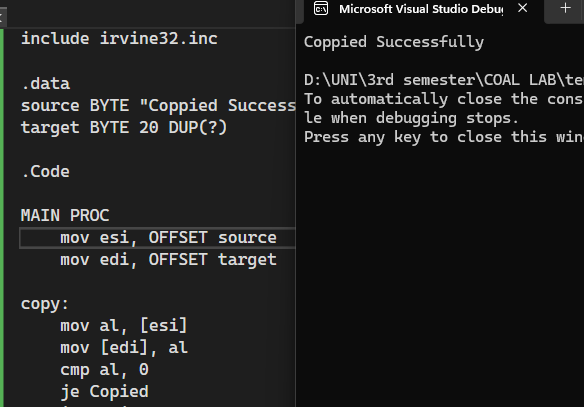
call WriteString

call crlf

EXIT

MAIN ENDP

END MAIN



TASK 05:  
include irvine32.inc

.data

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

.code

MAIN PROC

mov ecx, LENGTHOF array

mov esi, OFFSET array

mov edi, OFFSET array

add edi, TYPE array \* (LENGTHOF array - 1)

mov eax, 0

mov ebx, 0

mov edx, 0

loop1:

cmp esi, edi

jge endd

mov eax, [esi]

mov ebx, [edi]

mov [esi], ebx

mov [edi], eax

add esi, TYPE array

sub edi, TYPE array

loop loop1

endd:

mov ecx, LENGTHOF array

mov esi, OFFSET array

mov edx, 0

loop2:

mov eax, [esi]

call WriteDec

call crlf

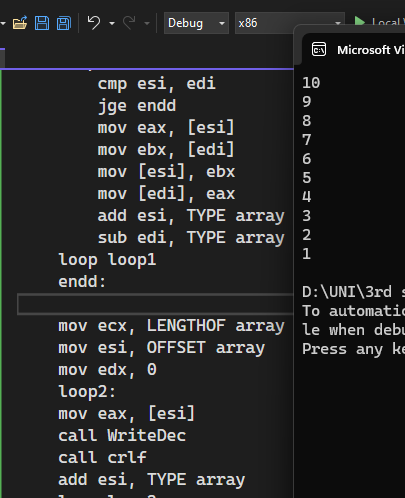
add esi, TYPE array

loop loop2

EXIT

MAIN ENDP

END MAIN



TASK 06:

include irvine32.inc

.data

array DWORD 8, 5, 1, 2, 6

arraySize DWORD 5

.code

main PROC

mov ecx, arraySize

dec ecx

outerLoop:

mov edx, ecx

mov esi, OFFSET array

innerLoop:

mov eax, [esi]

cmp eax, [esi+4]

jle nextPair

xchg eax, [esi+4]

mov [esi], eax

nextPair:

add esi, 4

loop innerLoop

mov ecx, edx

loop outerLoop

mov ecx, arraySize

mov esi, OFFSET array

displayLoop:

mov eax, [esi]

call Writedec

mov al, ' '

call WriteChar

add esi, 4

loop displayLoop

call Crlf

ret

EXIT

MAIN ENDP

END MAIN

