COAL LAB FINAL

Name: Ahmed Kasteer

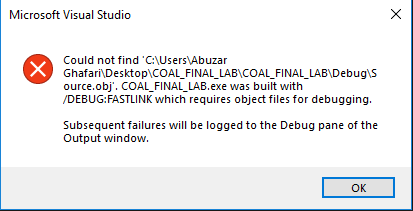
Section : 3D

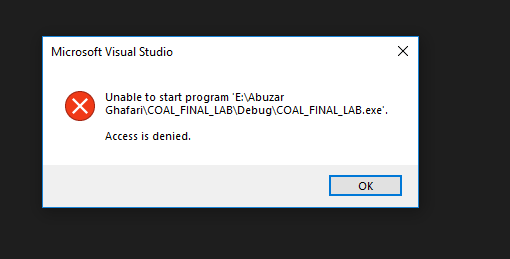
Roll Number: 20F-0336

2 Bonus Marks

My course instructor has taught me a lot other than just assembly language. Our instructor used to share thought of the day or a saying that used to be correct and very true and me along with my classmates realized that if we implicated this in our daily lives, we can be a lot better human being than we already are. Among some of those sayings were lessons about thankfulness, forgiveness, do not speak ill of others, always help each other and respect your teachers.

Respected Instructors. My code was not debugging due to some error I could not fix on my compiler. I asked the lab attendant to help me and he said the error is on your end and that the pc working fine. I tried explaining him but he denied and said it’s an error in your code which was not the case. I’ve done my work by honesty and you can consider my viva as well for lab. All my hard work went to waste just because of this. Kindly look into this matter as not even one of my codes was debugging. So, I couldn’t upload the screenshots of outputs as well. I tried my best to fix the error but couldn’t succeed and nor was I able to detect errors because of this. I leave my matter in your hands now.





Q1:

Part i)

INCLUDE Irvine32.inc

.data

X BYTE 0

.code

main PROC

mov ebx, 5

mov ecx, 7

cmp ebx, ecx

ja setX

setX:

mov X,1

call WaitMsg

exit

main ENDP

END main

Part ii)

INCLUDE Irvine32.inc

.data

X BYTE 0

.code

main PROC

mov edx, 5

mov ecx, 7

cmp ebx, ecx

ja else

jmp normal

else:

mov X,2

normal:

mov X, 1

call WaitMsg

exit

main ENDP

END main

Part iii)

INCLUDE Irvine32.inc

.data

X BYTE 0

val1 BYTE 0

.code

main PROC

mov val1 , 5

mov ecx, 7

mov edx, 8

cmp val1, ecx

ja nextcmp

nextcmp:

cmp ecx, edx

ja setvalue

mov X,2

setvalue:

mov X,1

call WaitMsg

exit

main ENDP

END main

Q2: ARRAY QUESTION

INCLUDE Irvine32.inc

input2darray PROTO, arraddr : PTR DWORD, rowsize : BYTE, colsize : BYTE

output2darray PROTO, arraddr :PTR DWORD, rowsize : BYTE, colsize : BYTE

sum2darray PROTO , arraddr: PTR DWORD, arr2addr : PTR DWORD, sumarraddr : PTR DWORD, rowsize : BYTE, colsize:BYTE

.data

arr1 BYTE 0,0,0

BYTE 0,0,0

BYTE 0,0,0

arr2 BYTE 0,0,0

BYTE 0,0,0

BYTE 0,0,0

summarr BYTE 0,0,0

BYTE 0,0,0

BYTE 0,0,0

row BYTE 0

col BYTE 0

ivar BYTE 0

jvar BYTE 0

prompt1 BYTE "Enter value [",0

prompt2 BYTE "][", 0

prompt3 BYTE "]:",0

.code

main PROC

INVOKE input2darray, ADDR arr1, 3,3

call crlf

INVOKE input2darray, ADDR arr2,3,3

call crlf

INVOKE output2darray, ADDR arr1,3,3

call crlf

INVOKE output2darray, ADDR arr2,3,3

call crlf

INVOKE sum2darray, ADDR arr1, ADDR arr2, ADDR summarr, 3, 3

INVOKE output2darray, ADDR summarr, 3,3

call crlf

call WaitMsg

exit

main ENDP

input2darray PROC, arraddr : PTR DWORD, rowsize : BYTE, colsize : BYTE

movzx esi, offset arraddr

movzx ecx, rowsize

mov ivar, 0

push ecx

loop1:

movzx ecx, colsize

mov jvar, 0

loop2:

movzx edx, prompt1

call writestring

movzx edx, ivar

call writedec

movzx edx, prompt2

call writestring

movzx edx, jvar

call writedec

movzx edx, prompt3

call writestring

call readdec

mov [esi], al

inc esi

inc jvar

loop loop2

pop ecx

call crlf

loop loop1

ret

input2darray endp

output2darray PROC, arraddr : PTR DWORD, rowsize : BYTE, colsize : BYTE

movzx esi, offset arraddr

movzx ecx, rowsize

push ecx

loop1:

movzx ecx, colsize

loop2:

mov edx, [esi]

call writedec

inc esi

loop loop2

pop ecx

call crlf

loop loop1

ret

output2darray ENDP

sum2darray PROC, arraddr : PTR DWORD, arr2addr : PTR DWORD, summarray : PTR DWORD,rowsize : BYTE, colsize : BYTE

mov esi, offset arraddr

mov edi, offset arr2addr

mov ebx, offset summarr

movzx ecx, rowsize

push ecx

loop1:

movzx ecx, colsize

loop2:

mov al, [esi]

add al, [edi]

mov [ebx], al

inc esi

inc edi

inc ebx

loop loop2

pop ecx

call crlf

loop loop1

ret

sum2darray ENDP

END main

Q3:

INCLUDE Irvine32.inc

.data

stones BYTE "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", 0

prompt1 BYTE "input name please", 0

prompt2 BYTE "input name2 please", 0

numstones BYTE "Enter num offset stones u want to remove please (min 1 and max 4)", 0

noStones BYTE ?

.code

main PROC

mov edx, offset stones

call writestring

mov edx, offset prompt1

call writestring

call readstring

mov eax, al

mov edx, offset prompt2

call writestring

call readstring

mov ebx, al

mov ecx, sizestones

loop1:

mov edx, offset numstones

call writestring

call readdec

mov edx,al

sub ecx, edx

call writestring

call readstring

mov ebx, al

call writestring

call readstring

mov ebx, al

loop loop1

call WaitMsg

exit

main ENDP

END main