COAL Lab 11 Assignment:

# Name: Owais Ali Khan

# Section: 3-F

# Roll no: 21K-3298

Question # 01:

INCLUDE Irvine32.inc

INCLUDE Macros.inc

.data

Str1 BYTE "127&j~3#^&\*#\*#45^", 0

.code

main PROC

push LENGTHOF Str1

push DWORD PTR Str1

call Scan\_String

exit

main ENDP

Scan\_String PROC

mov esi, 0

mov eax, '#'

mov ecx, LENGTHOF Str1

L1:

cmp Str1[esi], al

jz FOUND

inc esi

loop L1

mWrite "No instance of "

call WriteChar

mWrite " found in given string."

jmp endd

FOUND:

mWrite "Found instance of "

call WriteChar

mWrite " at index "

mov eax, esi

call WriteDec

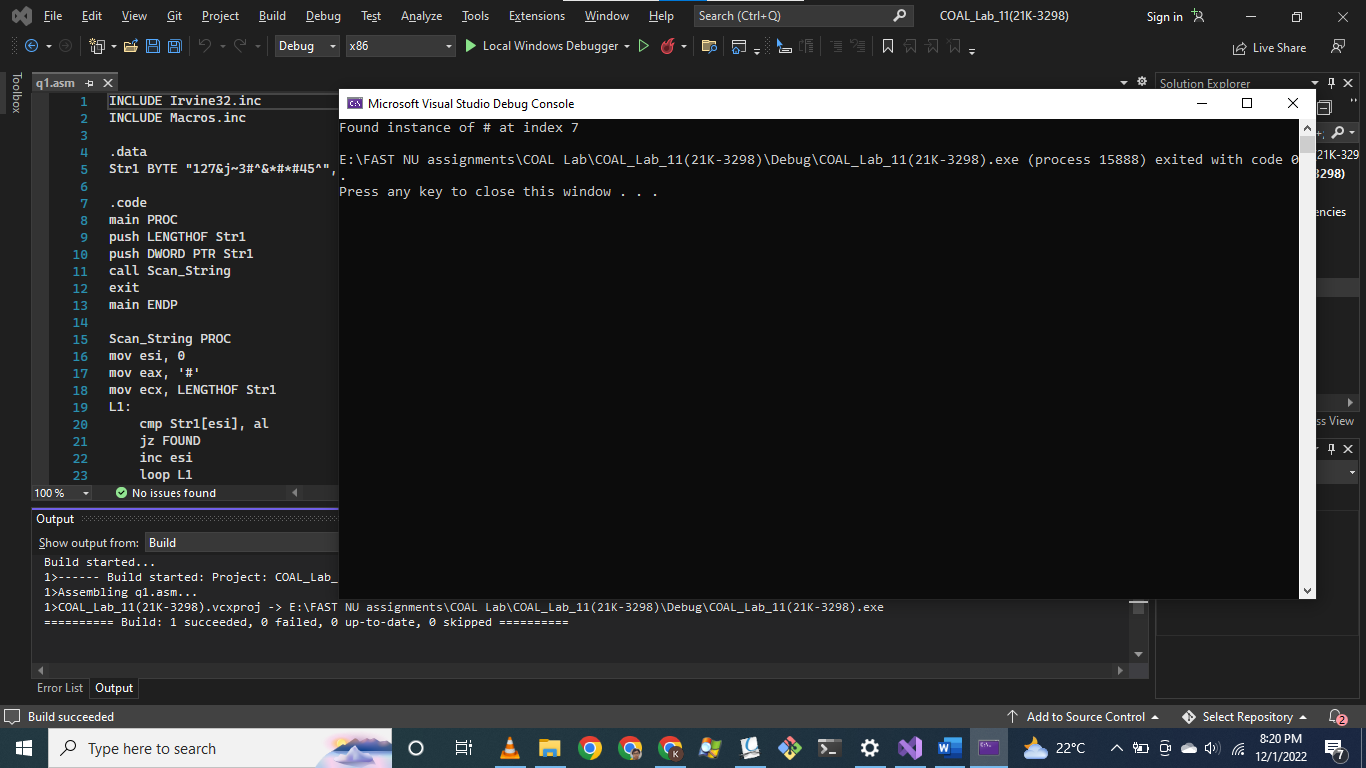
endd:

call Crlf

ret

Scan\_String ENDP

END main



Question # 02:

INCLUDE Irvine32.inc

INCLUDE Macros.inc

.data

beta byte "Not Found", 0

Str1 BYTE "127&j~3#^&\*#\*#45^", 0

.code

main PROC

push '#'

push LENGTHOF Str1

push OFFSET Str1

call Scan\_String

exit

main ENDP

Scan\_String PROC, string:DWORD, count:DWORD, character:DWORD

mov edi, string

mov ecx, count

;mov al, '#'

mov al, BYTE PTR character

repne scasb

jnz quit

found:

dec edi

mWrite "Found "

call WriteChar

mWrite " at index: "

sub edi, string

mov eax, edi

call WriteDec

jmp endd

quit:

mov edx, OFFSET beta

call WriteString

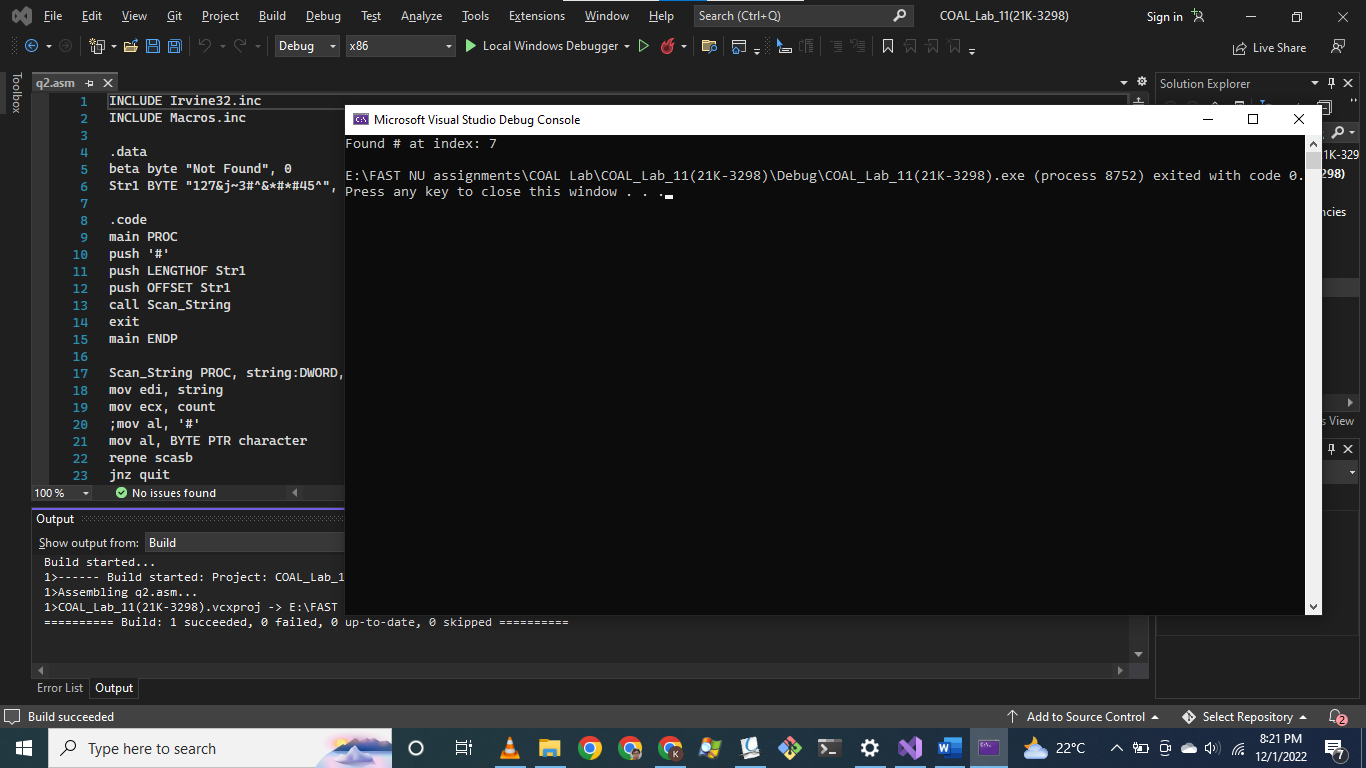
endd:

call Crlf

ret

Scan\_String ENDP

END main



Question # 03:

TITLE Question 6

INCLUDE Irvine32.inc

INCLUDE Macros.inc

.data

string1 BYTE "ABCDEFG", 0

string2 BYTE "ABCDEF", 0

.code

main PROC

IsCompare PROTO, source:PTR DWORD, target:PTR DWORD

INVOKE IsCompare, ADDR string1, ADDR string2

exit

main ENDP

IsCompare PROC, source:PTR DWORD, target:PTR DWORD

mWrite "First string: "

mov edx, source

call WriteString

call Crlf

mWrite "Second string: "

mov edx, target

call WriteString

call Crlf

call Crlf

INVOKE STR\_COMPARE, source, target

JA greater

JB lesser

JE equal

greater:

mWrite "First string is greater than second string."

jmp endd

lesser:

mWrite "First string is smaller than second string."

jmp endd

equal:

mWrite "Both strings are equal."

jmp endd

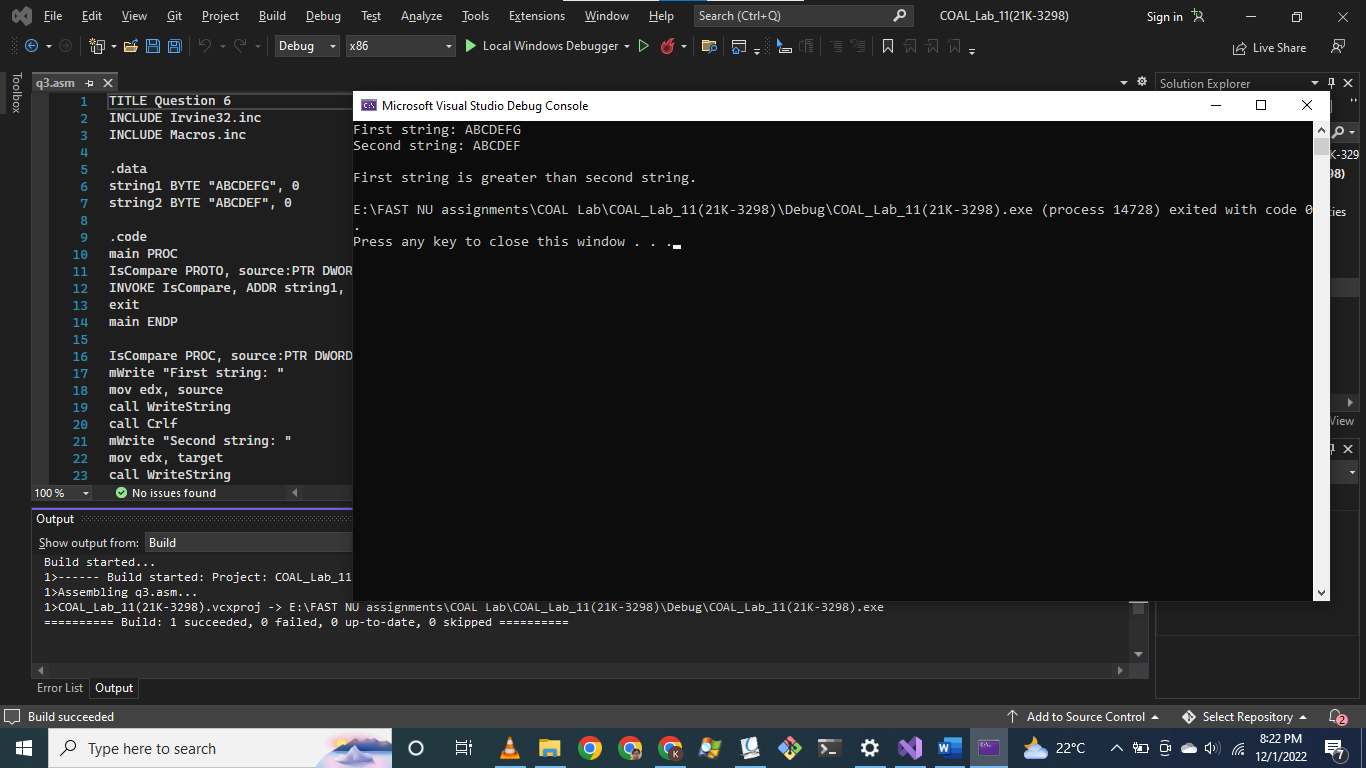
endd:

call Crlf

ret

IsCompare ENDP

END main



Question # 04:

TITLE Question 4

INCLUDE Irvine32.inc

INCLUDE Macros.inc

.data

source BYTE "Owais Ali Khan", 0

.code

main PROC

mWrite "Original: "

mov edx, OFFSET source

call WriteString

call Crlf

push LENGTHOF source - 1

push OFFSET source

call Str\_String

mWrite "Reversed: "

mov edx, OFFSET source

call WriteString

call Crlf

exit

main ENDP

Str\_String PROC ; Takes arguments: OFFSET string, length of string(without null terminator). Reverses in-place.

push ebp

mov ebp, esp

mov eax, 0

mov esi, [ebp + 8]

mov ecx, [ebp + 12]

L1:

mov al, [esi]

push eax

inc esi

loop L1

mov esi, [ebp + 8]

mov ecx, [ebp + 12]

L2:

pop eax

mov [esi], al

inc esi

loop L2

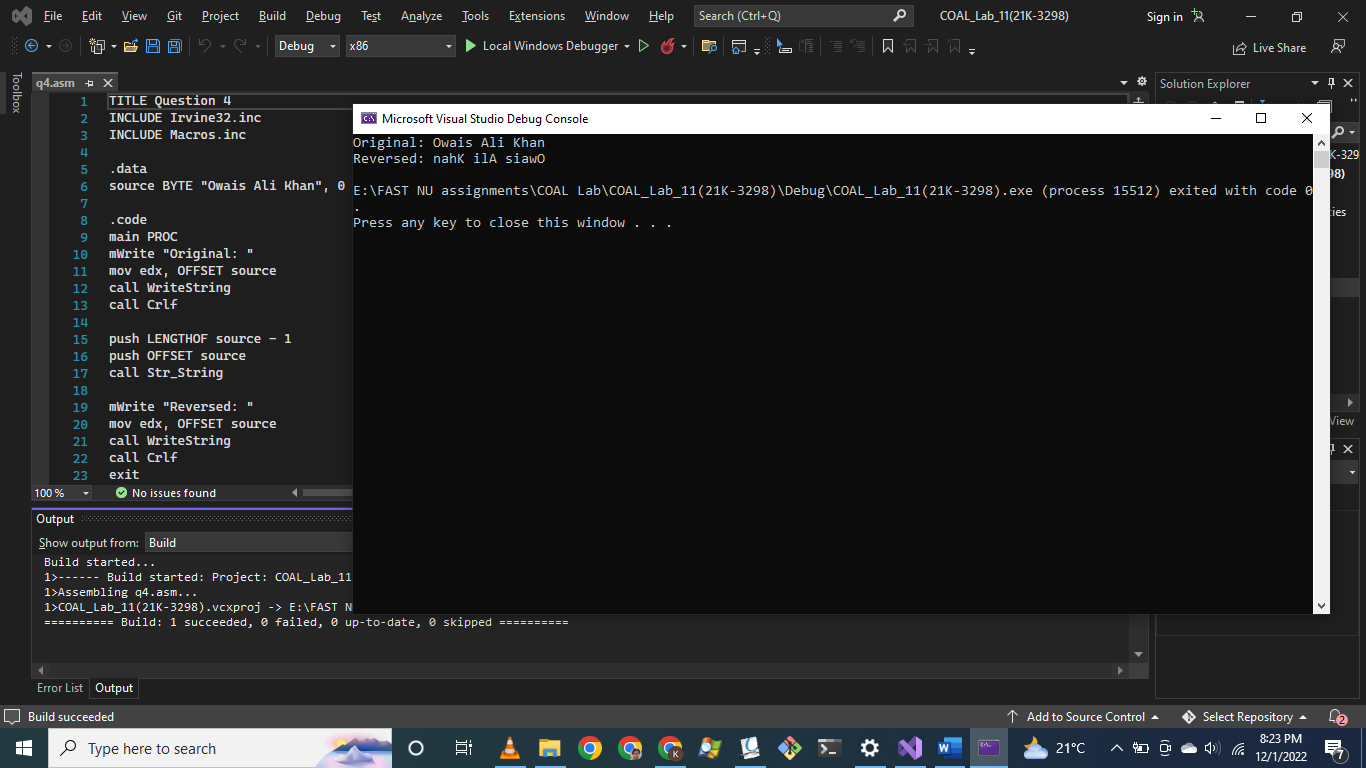
mov esp, ebp

pop ebp

ret

Str\_String ENDP

END main



Question # 05:

TITLE Question 5

INCLUDE Irvine32.inc

INCLUDE Macros.inc

.data

int\_array DWORD 10 DUP(5)

int\_array2 DWORD 10 DUP(0)

count DWORD 10

.code

main PROC

mWrite "Original arrays: "

call Crlf

push count

push OFFSET int\_array

call PRINT\_ARRAY

push count

call Crlf

push OFFSET int\_array2

call PRINT\_ARRAY

call Crlf

call Crlf

push count

push OFFSET int\_array2

push OFFSET int\_array

call Load

mWrite "Loaded arrays: "

call Crlf

push count

push OFFSET int\_array

call PRINT\_ARRAY

call Crlf

push count

push OFFSET int\_array2

call PRINT\_ARRAY

call Crlf

exit

main ENDP

Load PROC

push ebp

mov ebp, esp

mov esi, [ebp + 8]

mov edi, [ebp + 12]

mov ecx, [ebp + 16]

cld

L1:

lodsd

stosd

loop L1

mov esp, ebp

pop ebp

ret 12

Load ENDP

PRINT\_ARRAY PROC, array:DWORD, a\_length:DWORD

mov esi, array

mov ecx, a\_length

L1:

mov eax, [esi]

call WriteDec

mWrite " "

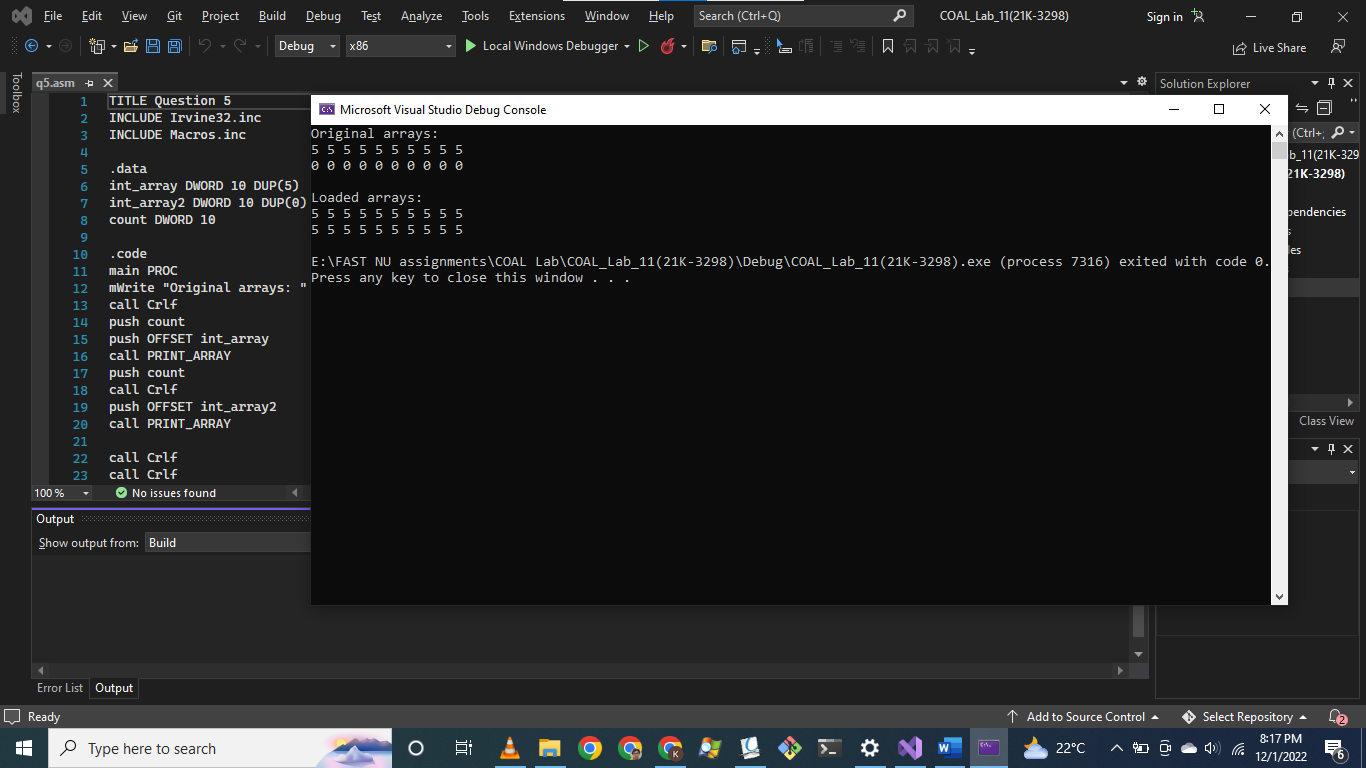
add esi, 4

loop L1

ret

PRINT\_ARRAY ENDP

END main



Question # 06:

TITLE Question 6

INCLUDE Irvine32.inc

INCLUDE Macros.inc

.data

global\_target BYTE "AAEBDCFBBC",0

global\_freqTable DWORD 256 DUP(0)

.code

main PROC

Get\_frequencies PROTO, target:PTR DWORD, freqTable:PTR DWORD

Print\_Frequencies PROTO, target:PTR DWORD, freqTable:PTR DWORD

INVOKE Get\_frequencies, ADDR global\_target, ADDR global\_freqTable

INVOKE Print\_frequencies, ADDR global\_target, ADDR global\_freqTable

exit

main ENDP

Get\_frequencies PROC, target:PTR DWORD, freqTable:PTR DWORD

mov esi, target

mov edi, freqTable

INVOKE STR\_LENGTH, target

mov ecx, eax

L1:

mov eax, 0

mov al, [esi]

inc DWORD PTR [edi + eax\*4]

inc esi

loop L1

ret

Get\_frequencies ENDP

Print\_Frequencies PROC, target:PTR DWORD, freqTable:PTR DWORD

mov esi, target

mov edi, freqTable

INVOKE STR\_LENGTH, target

mov ecx, eax

mWrite "Target String: "

L1:

mov eax, 0

mov al, [esi]

inc esi

call WriteChar

mWrite " "

loop L1

call Crlf

mWrite "ASCII Code: "

mov esi, target

INVOKE STR\_LENGTH, target

mov ecx, eax

L2:

mov eax, 0

mov al, [esi]

inc esi

call WriteDec

mWrite " "

loop L2

call Crlf

call Crlf

mWrite "Frequency Table: "

mov edi, freqTable

mov ecx, 256

L3:

mov eax, [edi]

cmp eax, 0

je zero\_freq

call WriteDec

mWrite " "

zero\_freq:

add edi, 4

loop L3

call Crlf

mWrite "Index: "

mov edi, freqTable

mov ecx, 256

L4:

mov eax, [edi]

cmp eax, 0

je zero\_freq2

mov eax, edi

sub eax, freqTable ; reverse of inc DWORD PTR [edi + eax\*4]

mov edx, 0

mov ebx, 4 ; Since freqTable is of type DWORD

div ebx

call WriteDec

mWrite " "

zero\_freq2:

add edi, 4

loop L4

call Crlf

ret

Print\_frequencies ENDP

END main

