23I-0544

MUHAMMAD HAMMAD

BCS-3D

LAB 12

TASK 01:

;1. Create a procedure named Scan\_String to find the index of the first occurrence of the character

;‘#’ in the given string.

;Str1 BYTE ‘127&j~3#^&\*#\*#45^’,0

INCLUDE Irvine32.inc

.data

string1 BYTE '127&j~3#^&\*#\*#45^',0

prompt1 BYTE "# found at index: ",0

prompt2 BYTE "# Not found",0

len DWORD 0

.code

main PROC

mov edi , OFFSET string1

mov ecx , LENGTHOF string1

mov len , ecx

mov al,'#'

call Scan\_String

exit

main ENDP

Scan\_String PROC

cld

repne scasb

jne notfound

mov eax,edi

sub eax,OFFSET string1

mov edx,OFFSET prompt1

call WriteString

call WriteDec

call Crlf

jmp quit

notfound:

mov edx,OFFSET prompt2

call WriteString

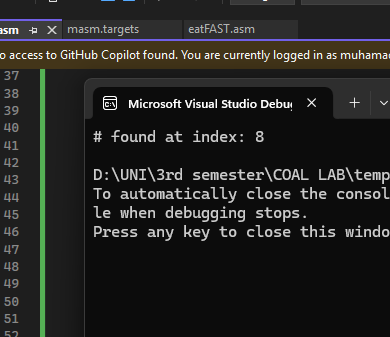
call Crlf

quit:

exit

Scan\_String ENDP

END main



Task 02:  
;1. Create a procedure named Scan\_String to find the index of the first occurrence of the character

;‘#’ in the given string.

;string1 BYTE ‘127&j~3#^&\*#\*#45^’,0

;2. Modify the above procedure to take offset of string1 and the character to be searched

;as argument.

INCLUDE Irvine32.inc

.data

string1 BYTE '127&j~3#^&\*#\*#45^',0

prompt1 BYTE "# found at index: ",0

prompt2 BYTE "# Not found",0

.code

main PROC

mov edi , OFFSET string1

mov al,'#'

movzx eax,al

push eax

push edi

call Scan\_String

exit

main ENDP

Scan\_String PROC

push ebp

mov ebp,esp

mov edi , [ebp+8]

mov al , [ebp+12]

mov ecx , LENGTHOF string1

cld

repne scasb

jnz notfound

mov eax,edi

sub eax,OFFSET string1

mov edx,OFFSET prompt1

call WriteString

call WriteDec

call Crlf

jmp quit

notfound:

mov edx,OFFSET prompt2

call WriteString

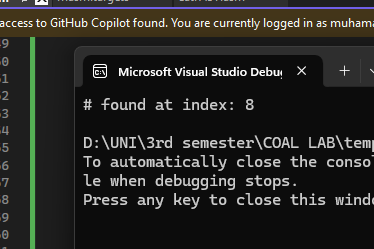
call Crlf

quit:

exit

Scan\_String ENDP

END main

  
  
Task 03:  
;3. Create IsCompare procedure to compare two strings.

INCLUDE Irvine32.inc

.data

string1 BYTE 'abcde',0

string2 BYTE 'abcdeee',0

promptG BYTE "String1 > String2",0

promptE BYTE "String1 = String2",0

promptL BYTE "String1 < String2",0

.code

main PROC

mov esi , OFFSET string1

mov edi , OFFSET string2

push esi

push edi

call Compare\_String

exit

main ENDP

Compare\_String PROC

push ebp

mov ebp,esp

mov esi , [ebp+12]

mov edi , [ebp+8]

repe cmpsb

JA greater

JB less

JZ equal

greater:

mov edx,OFFSET promptG

call WriteString

jmp quit

equal:

mov edx,OFFSET promptE

call WriteString

jmp quit

less:

mov edx,OFFSET promptL

call WriteString

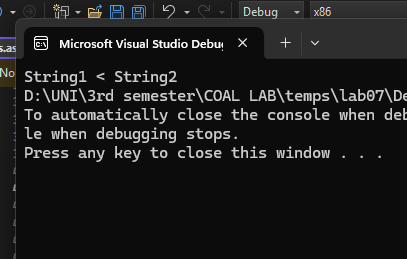
jmp quit

quit:

exit

Compare\_String ENDP

END main



TASK 04:

INCLUDE Irvine32.inc

.data

string1 BYTE "String Reversed", 0

Str\_Reverse PROTO, inputStr:DWORD

.code

main PROC

lea edx, string1

call WriteString

call Crlf

invoke Str\_Reverse, ADDR string1

call Crlf

lea edx, string1

call WriteString

call Crlf

exit

main ENDP

Str\_Reverse PROC, inputStr:DWORD

push esi

push edi

mov esi, inputStr

mov edi, inputStr

mov ecx, 0

L1:

cmp BYTE PTR [edi], 0

je Reverse

inc edi

inc ecx

jmp L1

Reverse:

dec edi

shr ecx, 1

RevLoop:

cmp esi, edi

jge Done

mov al, [esi]

xchg al, [edi]

mov [esi], al

inc esi

dec edi

loop RevLoop

Done:

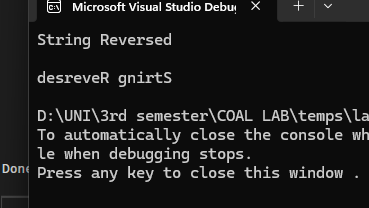
pop edi

pop esi

ret

Str\_Reverse ENDP

END main

  
TASK 05:

;5. Create a procedure that Loads an array of integer by multiplying it with

;Load(offset array, byte no)

INCLUDE Irvine32.inc

.data

arr1 DWORD 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

multiplierr DWORD 2

prompt1 BYTE "Array multipied",0

.code

main PROC

cld

mov esi , OFFSET arr1

mov edi , esi

mov ecx , LENGTHOF arr1

L1:

lodsd

mul multiplierr

stosd

loop L1

lea esi, arr1

mov ecx, LENGTHOF arr1

L2:

mov eax, [esi]

call Writedec

add esi, TYPE arr1

loop L2

call Crlf

call Crlf

lea edx , prompt1

call WriteString

call Crlf

exit

main ENDP

END main

