K230063

LAB 10

TASK 1

INCLUDE Irvine32.inc

.data

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

outputMsg BYTE "Hash symbol at position: "

.code

Find\_Hash PROC

mov edi, OFFSET inputStr

mov al, '#'

mov ecx, LENGTHOF inputStr

cld

repne scasb

jnz endProc

dec edi

mov ebx, LENGTHOF inputStr

sub ebx, ecx

mov eax, ebx

mov edx, OFFSET outputMsg

call crlf

call writeString

call writeDec

call crlf

endProc:

ret

Find\_Hash ENDP

start PROC

call Find\_Hash

Exit

start ENDP

END start

A screenshot of a computer program

AI-generated content may be incorrect.

TASK 2:

INCLUDE Irvine32.inc

.data

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

displayMsg BYTE "Hash found at position: ", 0

.code

start PROC

push LENGTHOF textInput

push OFFSET textInput

call Check\_Hash

exit

start ENDP

Check\_Hash PROC

push ebp

mov ebp, esp

pushad

mov edi, [ebp + 8]

mov ecx, [ebp + 12]

mov al, '#'

cld

repne scasb

jnz skipPrint

dec edi

sub edi, [ebp + 8]

mov eax, edi

mov edx, OFFSET displayMsg

call crlf

call WriteString

call WriteDec

call crlf

skipPrint:

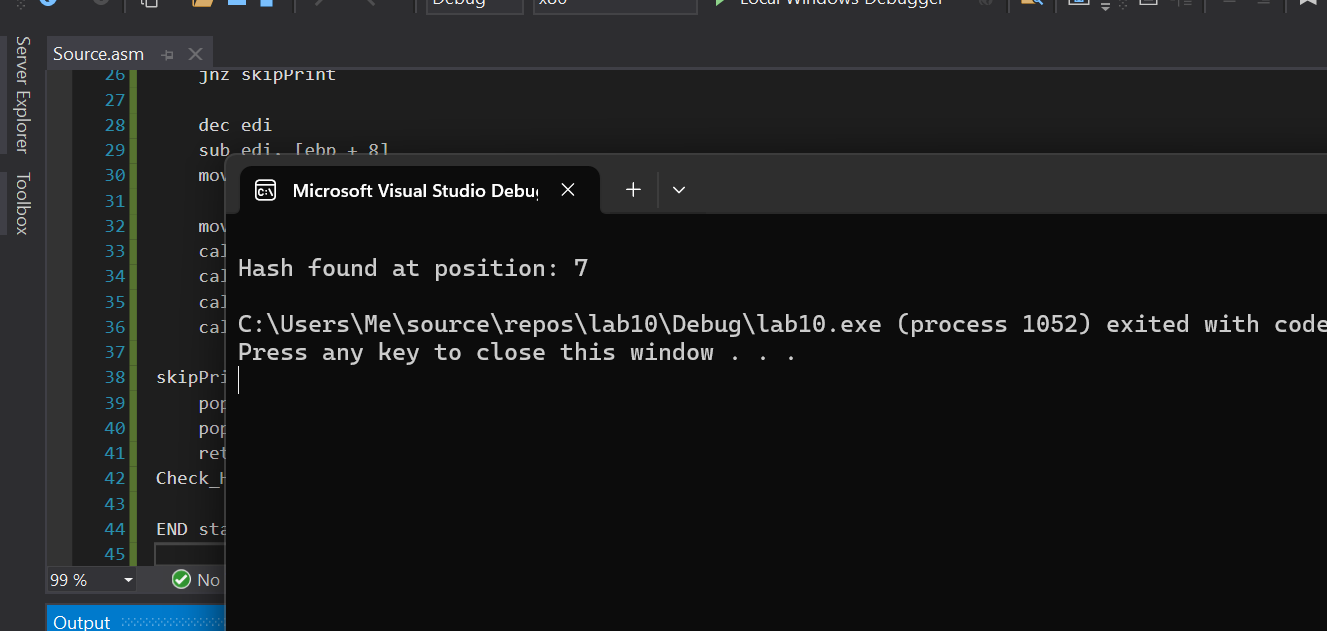
popad

pop ebp

ret 8

Check\_Hash ENDP

END start



TASK 3:

INCLUDE Irvine32.inc

.data

inputA BYTE 'ghijk', 0

inputB BYTE 'abcd', 0

msgHigh BYTE 'inputA > inputB', 0

msgLow BYTE 'inputA < inputB', 0

msgEqual BYTE 'inputA == inputB', 0

.code

main PROC

push OFFSET inputB

push OFFSET inputA

call CompareStrings

exit

main ENDP

CompareStrings PROC

push ebp

mov ebp, esp

pushad

mov esi, [ebp + 12]

mov edi, [ebp + 8]

checkNext:

lodsb

scasb

jne decide

cmp al, 0

jne checkNext

mov edx, OFFSET msgEqual

jmp show

decide:

ja showGreater

jb showLess

showGreater:

mov edx, OFFSET msgHigh

jmp show

showLess:

mov edx, OFFSET msgLow

show:

call crlf

call WriteString

call crlf

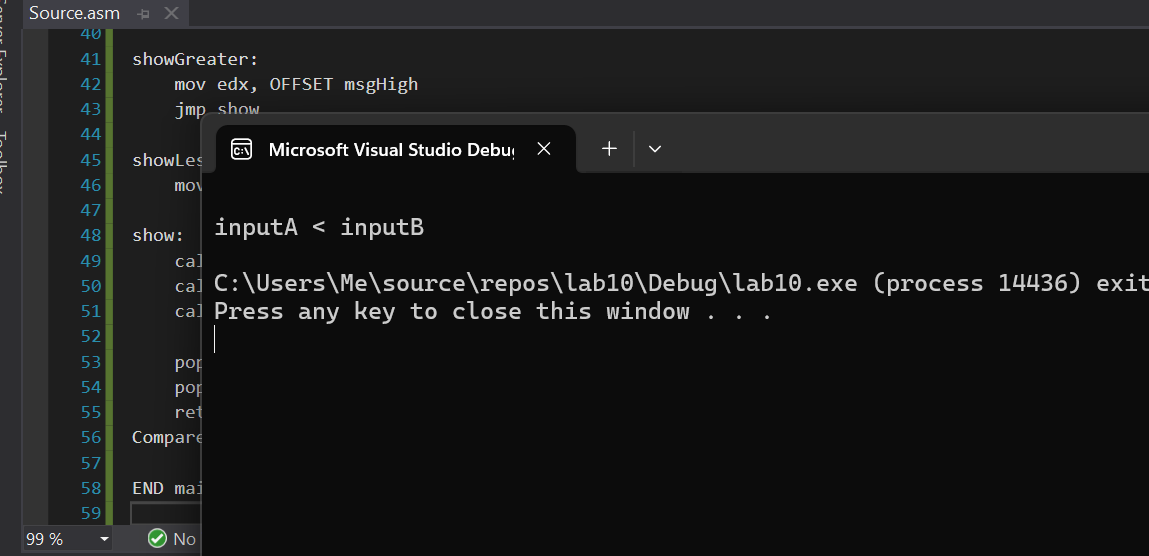
popad

pop ebp

ret 8

CompareStrings ENDP

END main



TASK 4:

INCLUDE Irvine32.inc

.data

original BYTE "hello world", 0

flipped BYTE LENGTHOF original DUP(0)

.code

main PROC

push OFFSET original

push LENGTHOF original

call ReverseChars

exit

main ENDP

ReverseChars PROC

push ebp

mov ebp, esp

pushad

mov esi, [ebp + 8] ; OFFSET original

mov ecx, [ebp + 12] ; LENGTHOF original

mov edi, OFFSET flipped

add esi, ecx

dec esi

std

reverseLoop:

lodsb

stosb

loop reverseLoop

cld

mov byte ptr [edi], 0 ; null-terminate

mov edx, OFFSET flipped

call crlf

call WriteString

call crlf

popad

pop ebp

ret 8

ReverseChars ENDP

END main

A computer screen with white text

AI-generated content may be incorrect.

TASK 5:

INCLUDE Irvine32.inc

.data

arr DWORD 1,2,3,4,5

factor DWORD 20

.code

main PROC

push OFFSET arr

push LENGTHOF arr

push TYPE arr

push OFFSET factor

call ProcessArray

exit

main ENDP

ProcessArray PROC

push ebp

mov ebp, esp

mov esi, [ebp + 20]

mov ecx, [ebp + 16]

mov ebx, [ebp + 8]

mov edi, esi

cld

LoopStart:

lodsd

mul ebx

stosd

loop LoopStart

mov esi, [ebp + 20]

mov ecx, [ebp + 16]

mov ebx, [ebp + 12]

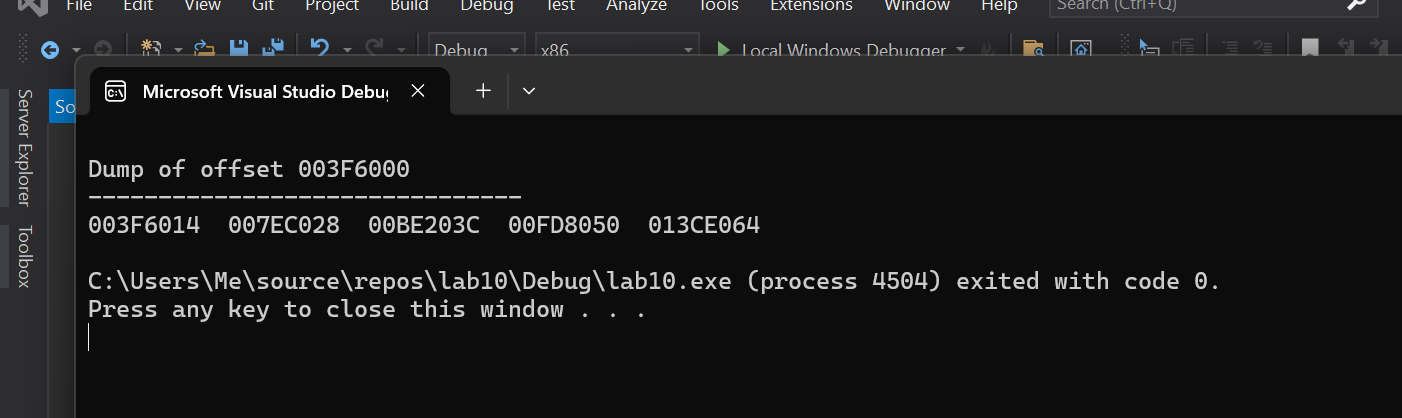
call dumpmem

pop ebp

ret

ProcessArray ENDP

END main



TASK 6:

INCLUDE Irvine32.inc

.data

target BYTE "AAEBDCFBBC"

freqTable DWORD 256 DUP(0)

.code

Get\_frequencies PROC

push ebp

mov ebp, esp

mov esi, [ebp + 8] ; pfreqTable

mov edi, [ebp + 12] ; ptarget

mov ecx, LENGTHOF target

L1:

movzx eax, BYTE PTR [edi]

mov ebx, 4

mul ebx

add esi, eax

add DWORD PTR [esi], 1

inc edi

loop L1

mov esi, [ebp + 8]

mov ecx, 256

call crlf

L2:

mov eax, [esi]

cmp eax, 0

je \_continue

call writedec

call crlf

\_continue:

add esi, 4

loop L2

pop ebp

ret

Get\_frequencies ENDP

main PROC

invoke Get\_frequencies, OFFSET target, OFFSET freqTable

exit

main ENDP

END main

