**COAL**

**LAB 11**

**21k-4834**

**TASK 1**

Include Irvine32.inc

.data

Str1 BYTE "8142t&f!3#\_&(#&#435^",0

prompt BYTE "# is on: ",0

.code

main PROC

call Scan\_String

exit

main endp

Scan\_String PROC

mov edi,OFFSET Str1

mov al,'#'

mov ecx,LENGTHOF Str1

repne scasb

jnz quit

sub edi,OFFSET Str1

Graphical user interface, text

Description automatically generateddec edi

mov eax,edi

mov edx,OFFSET prompt

call writestring

call writedec

ret

quit:

exit

Scan\_String endp

end main

END main

**TASK 2**

include Irvine32.inc

.data

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

prompt byte "The Index of # is: ",0

.code

main PROC

mov edx,OFFSET Str1

call Scan\_String

exit

main endp

Scan\_String PROC

mov edi,edx

mov al,[ebx]

A screenshot of a computer

Description automatically generated with medium confidencemov ecx,LENGTHOF Str1

cld

repne scasb

jnz quit

sub edi,OFFSET Str1

dec edi

mov eax,edi

mov edx,OFFSET prompt

call writestring

call writedec

ret

quit:

exit

Scan\_String endp

END main

**TASK 3:**

include Irvine32.inc

.data

string1 byte 'Toyota' ,0

string2 byte 'Revo',0

prompt1 byte 'String1 Is Greater than String2',0

prompt2 byte 'String1 Is Lesser than String2',0

.code

main PROC

call IsCompare

exit

main ENDP

IsCompare PROC

mov esi,OFFSET string1

mov edi, OFFSET string2

mov ecx,LENGTHOF string1

cmpsd

ja L1

mov edx,OFFSET prompt2

call WriteString

exit

L1:

mov edx,OFFSET prompt1

call WriteString

ret

IsCompare endp

END main



**TASK 4:**

include Irvine32.inc

.data

str1 byte "I want to earn money",0

.code

main PROC

mov ebx,lengthof str1 - type str1

call Str\_Reverse

exit

main ENDP

Str\_Reverse PROC

mov al,[str1+ebx]

call writechar

dec ebx

cmp ebx,0

jl rec\_END

call Str\_Reverse

rec\_END:

ret

Str\_Reverse ENDP

END main

**TASK 5**

include Irvine32.inc

.data

arr byte 1,2,3,4,5

space byte ' ',0

text1 byte "ARRAY BEFORE MULTIPLYING",0Dh,0Ah,0

text2 byte "ARRAY AFTER MULTIPLYING",0Dh,0Ah,0

no byte 2

.code

main PROC

mov esi,OFFSET arr

mov ecx,LENGTHOF arr

mov edi,0

Text

Description automatically generatedmov edx,OFFSET text1

call WriteString

L2:

mov eax,0

mov al,[esi+edi]

call WriteDec

mov edx,OFFSET space

call WriteString

inc edi

loop L2

mov esi,OFFSET arr

call \_LoadFunc

mov edi,0

mov esi,OFFSET arr

call crlf

mov ecx,LENGTHOF arr

mov edx,OFFSET text2

call WriteString

L3:

mov eax,0

mov al,[edi+esi]

call WriteDec

mov edx,OFFSET space

call WriteString

inc edi

loop L3

exit

main endp

\_LoadFunc PROC

mov ecx,LENGTHOF arr

mov edi,0

L1:

mov eax,0

mov al,[edi+esi]

mul no

mov [esi+edi],al

inc edi

loop L1

ret

\_LoadFunc endp

END main

**TASK 6**

include Irvine32.inc

.data

target byte "AAEBDCFBBC",0

freqTable dword 256 DUP(0)

eql byte '=',0

var byte 65d

.code

Get\_frequencies PROC targets: PTR dword, freqT: PTR dword

mov ebp, esp

mov esi, targets

mov ecx, esi

mov esi, freqT

L1:

mov ebx, [ecx]

movzx eax, bl

cmp eax,0

je done

mov edx, 4

imul edx

mov edx, esi

add edx, eax

mov eax, [edx]

inc eax

mov [edx], eax

inc ecx

jmp L1

done:

ret

Get\_frequencies ENDP

main PROC

mov esi, offset target

mov esi, offset freqTable

Get\_frequencies PROTO, targ: PTR dword, freq: PTR dword

INVOKE Get\_frequencies, ADDR target, ADDR freqTable

movzx eax, target[3]

mov ebx, 4

imul ebx

mov ebx, eax

mov eax, freqTable[ebx]

call crlf

mov esi,OFFSET freqTable

mov ecx,LENGTHOF freqTable

mov bl,00

l1:

mov al,bl

cmp al,var

jne L2

call writechar

mov eax, [esi]

mov edx,OFFSET eql

call writestring

call writedec

add var,1

call crlf

L2:

add esi,4

inc bl

loop l1

main ENDP

exit

END main