**COAL LAB 11**

Q.1)

INCLUDE irvine32.inc

.data

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

msg byte "Index: ",0

.code

main PROC

call Scan\_String

mov edx,offset msg

call writestring

call writedec

call crlf

exit

main endp

Scan\_String proc

mov eax,0

mov ebx,0

mov edi,offset Str1

mov al,"#"

mov ecx,lengthof str1

L1:

cld

scasb [edi]

je return

inc ebx

loop L1

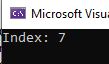
return:

mov eax,ebx

ret

Scan\_String endp

end main



Q.2)

INCLUDE irvine32.inc

Scan\_String proto ,ptrSt:ptr byte,char:byte

.data

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

msg byte "Index: ",0

.code

main PROC

INVOKE Scan\_String,offset str1,"#"

mov edx,offset msg

call writestring

call writedec

call crlf

exit

main endp

Scan\_String proc ,ptrSt:ptr byte,char:byte

mov eax,0

mov ebx,0

mov edi,ptrSt

mov al,char

mov ecx,lengthof str1

L1:

cld

scasb [edi]

je return

inc ebx

loop L1

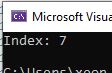
return:

mov eax,ebx

ret

Scan\_String endp

end main



Q.3)

INCLUDE irvine32.inc

.data

str1 BYTE "abc",0

str2 BYTE "abcd",0

msg1 byte "String 1 is greater ",0

msg2 byte "String 2 is greater ",0

msg3 byte "Strings are equal ",0

.code

main PROC

mov esi,0

mov edi,0

mov esi,offset str1

mov edi,offset str2

cmpsd

ja l1

cmpsd

jb l2

mov edx,offset msg3

call writestring

jmp exitt

l1:

mov edx,offset msg1

call writestring

jmp exitt

l2:

mov edx,offset msg2

call writestring

exitt:

call crlf

exit

main endp

end main



Q.4)

INCLUDE Irvine32.inc

.data

text BYTE "abcdef",0

msg BYTE "Reverse string: ",0

.code

main PROC

mov ebx,sizeof text - type text

mov edx,offset msg

call writestring

call Str\_Reverse

call crlf

exit

main ENDP

Str\_Reverse PROC

mov al,[text+ebx]

call writechar

dec ebx

cmp ebx,0

jl rett

call Str\_Reverse

rett:

ret

Str\_Reverse ENDP

END main



Q.5)

INCLUDE Irvine32.inc

.data

arr BYTE 1,2,3,4,5

msg BYTE "AFTER MULTIPLICATION:",0

comma BYTE ',',0

no BYTE 3

.code

main PROC

mov esi,OFFSET arr

mov ecx,LENGTHOF arr

mov edi,0

L2:

mov eax,0

mov al,[esi+edi]

call WriteDec

mov edx,OFFSET comma

call WriteString

inc edi

loop L2

mov esi,OFFSET arr

call Load\_

mov edi,0

mov esi,OFFSET arr

call crlf

mov edx ,offset msg

call writestring

call crlf

mov ecx,LENGTHOF arr

L3:

mov eax,0

mov al,[edi+esi]

call WriteDec

mov edx,OFFSET comma

call WriteString

inc edi

loop L3

exit

main endp

Load\_ PROC

mov ecx,sizeof arr

mov edi,0

L1:

mov eax,0

mov al,[edi+esi]

mul no

mov [esi+edi],al

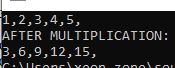
inc edi

loop L1

ret

Load\_ endp

END main



Q.6)

INCLUDE Irvine32.inc

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

.data

target BYTE "AAEBDCFBBC",0

freqTable DWORD 256 DUP(0)

space byte ' ',0

var byte 65d

.code

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,0

mov eax, freqTable[ebx]

mov esi,OFFSET freqTable

mov ecx,LENGTHOF freqTable

L1:

mov al,bl

cmp al,var

jne L2

call writechar

mov eax, [esi]

mov edx,OFFSET space

call writestring

call writedec

inc var

call crlf

L2:

add esi,4

inc bl

loop L1

main ENDP

exit

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 return

mov edx, 4

mul edx

mov edx, esi

add edx, eax

mov eax, [edx]

inc eax

mov [edx], eax

inc ecx

jmp L1

return:

ret

Get\_frequencies ENDP

END main

