**Name: Mozeb Ahmed Khan**

**Roll No: 20F-0161**

**Sec: CS-3A**

**Subj: Coal**

**Assignment No: 04**

**Q1:**

**Code:**

INCLUDE Irvine32.inc

.data

Msg1 byte "Enter hexadecimal value from 0-F: ", 0

Msg2 byte "The equivalent decimal value is ", 0

Msg3 byte "Invalid input.", 0

Value byte ?

.code

main PROC

call InputFunc

cmp ebx, 1

je Error

call writechar

call ConvertFunc

Error :

call crlf

exit

main endp

InputFunc proc

mov eax, 0

mov edx, offset Msg1

call writestring

call readchar

cmp al, 0

jng Invalid

cmp al, 'F'

ja Invalid

mov Value, al

jmp Valid

Invalid :

call writechar

mov ebx, 1

call crlf

mov edx, offset Msg3

call writestring

ret

Valid :

mov ebx, 2

ret

InputFunc endp

ConvertFunc proc

cmp Value, '9'

jbe Decimal

cmp Value, 'F'

jmp Hexadecimal

Decimal :

sub al, 30h

mov Value, al

mov eax, 0

mov al, Value

call OutputFunc

ret

Hexadecimal :

sub al, 51h

call OutputFunc

ret

ConvertFunc endp

OutputFunc proc

call crlf

mov edx, offset Msg2

call writestring

mov Value, al

mov eax, 0

mov al, Value

call writehex

ret

OutputFunc endp

END main

**Screenshot:**

Text

Description automatically generated

**Q2:**

**Code:**

INCLUDE Irvine32.inc

.data

Val1 dword 150d

Val2 dword 4d

Result dword ?

Msg byte "The Bitwise Multiplication is: ", 0

.code

main PROC

call MulFunc

exit

main ENDP

MulFunc proc

mov edi, 0

mov ebx, Val1

mov ecx, 32

l1:

push ecx

mov ecx, edi

clc

shr ebx, 1

jnc l2

mov eax, Val2

shl eax, cl

add Result, eax

l2 :

inc edi

pop ecx

loop l1

mov eax, Result

mov edx, offset Msg

call writestring

call writedec

ret

MulFunc endp

END main

**Screenshot:**

Text

Description automatically generated with low confidence

Q3:

Code:

INCLUDE Irvine32.inc

.data

X qword 20403375962047A1h

Y qword 55210304A22A7B2h

.code

main PROC

mov ecx, 8

mov esi, offset X

mov edi, offset Y

clc

l1 :

mov al, byte ptr[esi]

sbb al, byte ptr[edi]

mov byte ptr[esi], al

inc esi

inc edi

loop l1

exit

main endp

END main

Output:

Q5

Code:

INCLUDE irvine32.inc

.data

Arr DWORD 1, 3, 5, 7, 9

X DWORD 3

.code

main PROC

cld

mov esi, OFFSET Arr

mov edi, esi

mov ecx, LENGTHOF Arr

l1 :

lodsd

mul X

stosd

loop l1

call waitmsg

exit

main ENDP

END main

Output:

Q6:

INCLUDE Irvine32.inc

String\_Length proto,

Target\_String: ptr byte

String\_Find proto, Target\_String : ptr byte, SourceStr : ptr byte

.data

target BYTE 20 dup(0)

source BYTE 20 dup(0)

posOffset dword ?

pos DWORD ?

dummy byte ?

str1 byte "Input target string: ", 0

str2 byte "Input source string: ", 0

msg1 byte "Target Found", 0

msg2 byte "Offset is: ", 0

msg3 byte "Index is: ", 0

msg4 byte "Target Not found", 0

.code

main PROC

or al, 1

mov edx, offset str1

call writestring

mov edx, offset target

mov ecx, sizeof target

call readstring

call crlf

mov edx, offset str2

call writestring

mov edx, offset source

mov ecx, sizeof source

call readstring

invoke String\_Length, addr target

invoke String\_Find, addr target, addr source

cmp posOffset, 0

je notfound

mov edx, offset msg1

call WriteString

call Crlf

mov edx, offset msg2

call WriteString

mov eax, posOffset

call WriteHex

call Crlf

mov edx, offset msg3

call WriteString

mov eax, pos

call WriteDec

call Crlf

jmp Ending

notFound :

mov edx, offset msg4

call WriteString

call Crlf

Ending :

main ENDP

String\_Length proc uses esi eax,

Target\_String : ptr byte

mov esi, Target\_String

mov ecx, 0

L1 :

mov al, [esi]

cmp al, 0

je L2

inc ecx

inc esi

jmp L1

L2 :

ret

String\_Length endp

String\_Find proc uses esi edi edx ebx eax ecx, Target\_String : ptr byte, SourceStr : ptr byte

mov esi, Target\_String

L5 :

mov edx, esi

mov edi, SourceStr

L8 :

mov al, [edx]

mov bl, [edi]

cmp bl, 0

je L6

cmp al, bl

jne L7

inc edx

inc edi

jmp L8

L6 :

mov posOffset, esi

mov eax, esi

test dl, 0

jmp L9

L7 :

inc esi

or al, 1

loop L5

L9 :

mov esi, Target\_String

sub eax, esi

mov pos, eax

ret

String\_Find endp

END main