**Coal Assignment 4**

*Ahmed Kasteer*

*20F-0336*

*Section 3D*

Question 1

INCLUDE Irvine32.inc

.data

PROMPT BYTE "Input Hexadecimal digit: ",0

DECIMAL BYTE "Decimal digit: ",0

.code

InputProcedure PROC

mov edx,offset PROMPT

call WriteString

call ReadChar

mov bl,al

InputProcedure ENDP

END InputProcedure

convertDecimalProcedure PROC

cmp bl,'A'

jb less\_than\_10

jmp greater\_than\_9

less\_than\_10:

mov edx,offset DECIMAL

call WriteString

mov al,bl

call WriteChar

jmp done

greater\_than\_9:

mov edx,OFFSET DECIMAL

call WriteString

cmp al,'A'

jz A

jnz else1

A:

mov eax,10

call WriteDec

jmp done

else1:

cmp al,'B'

jz B

JNZ else2

B:

mov eax,11

call WriteDec

jmp done

else2:

cmp al,'C'

jz true1

jnz else3

true1:

mov eax,12

call WriteDec

jmp done

else3:

cmp al,'D'

jz D

jnz else4

D:

mov ax,13

call WriteDec

jmp done

else4:

cmp al,'E'

jz E

jnz else5

E:

mov eax,14

call WriteDec

jmp done

else5:

cmp al,'F'

jz F

F:

mov eax,15

call WriteDec

convertDecimalProcedure ENDP

END convertDecimalProcedure

main PROC

call InputProcedure

call convertDecimalProcedure

exit

main ENDP

END main

Text

Description automatically generated

Question 2

INCLUDE Irvine32.inc

.data

var1 BYTE "Enter number: ",0

var2 BYTE "Enter number to multiply with: ",0

var3 BYTE "Result: ",0

.code

main PROC

mov edx, OFFSET var1

call WriteString

call ReadDec

mov ebx, eax

mov edx, OFFSET var2

call WriteString

call ReadDec

call Multiply

mov edx,offset var3

call WriteString

call WriteDec

exit

main ENDP

Multiply PROC USES EBX ECX EDX

mov edx, eax

xor eax, eax

bsr ecx, edx

L1:

shr edx, 1

jnc @F

add eax, ebx

@@:

shl ebx, 1

sub ecx, 1

jnc L1

ret

Multiply ENDP

END main

Text

Description automatically generated

Question 3

INCLUDE Irvine32.inc

.data

val1 QWORD 20403004362047A1h

val2 QWORD 055210304A2630B2h

r QWORD 0

.code

main PROC

mov ecx, 8

mov esi , offset val1

mov edi , offset val2

mov ebx , offset r

clc

top:

mov al, BYTE PTR[esi]

sub al, BYTE PTR[edi]

mov BYTE PTR [ebx], al

inc esi

inc edi

inc ebx

loop top

exit

main ENDP

END main

Question 5

INCLUDE Irvine32.inc

.data

arr1 DWORD 2,4,6,8,10

const DWORD 4

.code

main PROC

cld

mov esi,offset arr1

mov edi,esi

mov ecx,LENGTHOF arr1

L1:

lodsd

mul const

stosd

LOOP L1

mov esi,offset arr1

mov edi,esi

mov ecx,LENGTHOF arr1

L2:

mov eax,esi

call WriteDec

loop L2

exit

main ENDP

END main

Text

Description automatically generated

Question 6

INCLUDE Irvine32.inc

find PROTO, pTarget:PTR BYTE, pSource:PTR BYTE

.data

target BYTE "1230987abcgd",0

source BYTE "987",0

Display1 BYTE "String found at position: ",0

Display3 BYTE "Unable to find Source string in the target.",0Ah,0Ah,0Dh,0

stop DWORD ?

lenTarget DWORD ?

lenSource DWORD ?

position DWORD ?

.code

main PROC

INVOKE find,ADDR target, ADDR source

mov position,eax

jz found1

mov edx,OFFSET Display3

call WriteString

jmp quit

found1:

mov edx,OFFSET Display1

call WriteString

mov eax,position

call WriteDec

quit:

exit

main ENDP

find PROC, pTarget:PTR BYTE,pSource:PTR BYTE

INVOKE Str\_length,pTarget

mov lenTarget,eax

INVOKE Str\_length,pSource

mov lenSource,eax

mov edi,OFFSET target

mov esi,OFFSET source

mov eax,edi

add eax,lenTarget

sub eax,lenSource

inc eax

mov stop,eax

cld

mov ecx,lenSource

L1:

pushad

repe cmpsb

popad

je found

inc edi

cmp edi,stop\_pos

jae notfound

jmp L1

notfound:

or eax,1

jmp done

found:

mov eax,edi

sub eax,pTarget

cmp eax,eax

done:

ret

find ENDP

END main

Text

Description automatically generated