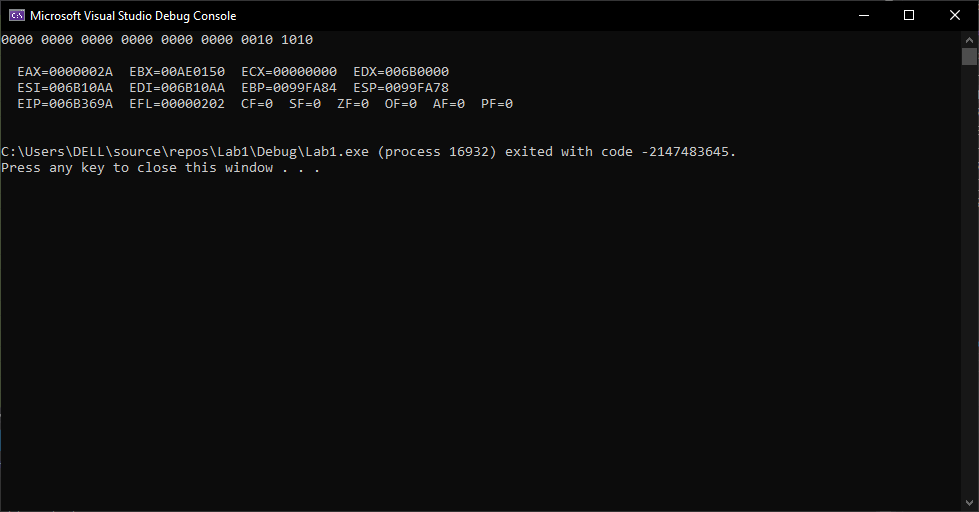
**COAL LAB 8**

**SYED YOUSHA**

**BSR-3C**

**K226007**

Task1:



TITLE ReverseArray.asm

INCLUDE Irvine32.inc

.data

multiplicant word 10101b

multiplier word 10b

result word 0

.code

main PROC

mov ecx, 4

mov bx, multiplicant

mov dx, multiplier

l1:

shr dx, 1

jnc skip

add result, bx

skip:

shl bx, 1

loop l1

movzx eax, result

call writebin

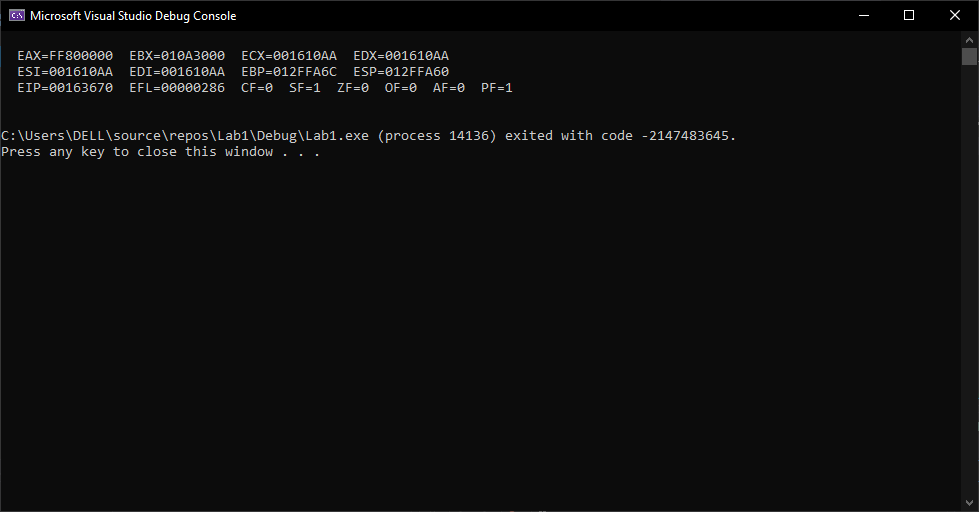
call crlf

call dumpregs

main ENDP

END main

Task2:



TITLE ReverseArray.asm

INCLUDE Irvine32.inc

.data

.code

main PROC

mov ax, -128

shl eax, 16

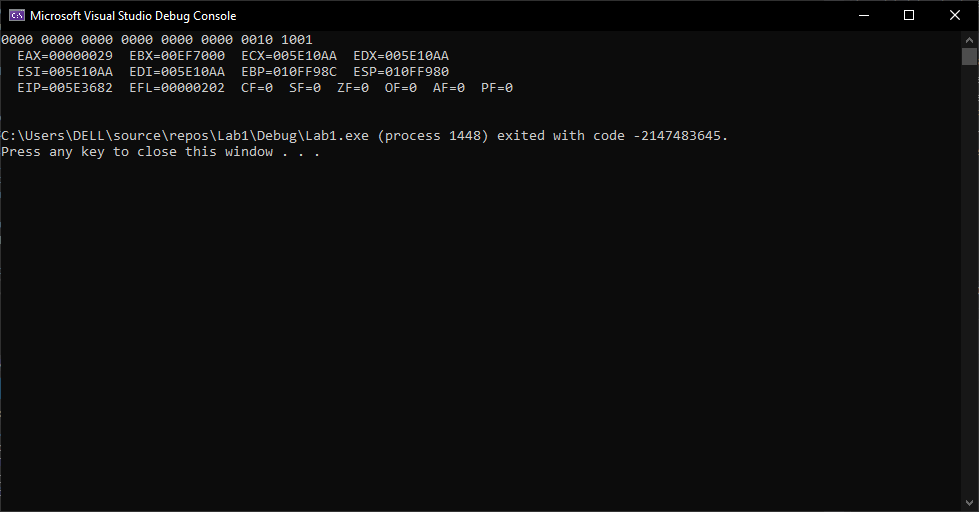
ror ax, 16

call dumpregs

main ENDP

END main

Task3:



TITLE ReverseArray.asm

INCLUDE Irvine32.inc

.data

ts word 1101101010011010b

bmins word 0

.code

main PROC

movzx eax, ts

shr eax, 4

shl ax, 10

shr ax, 10

mov bmins, ax

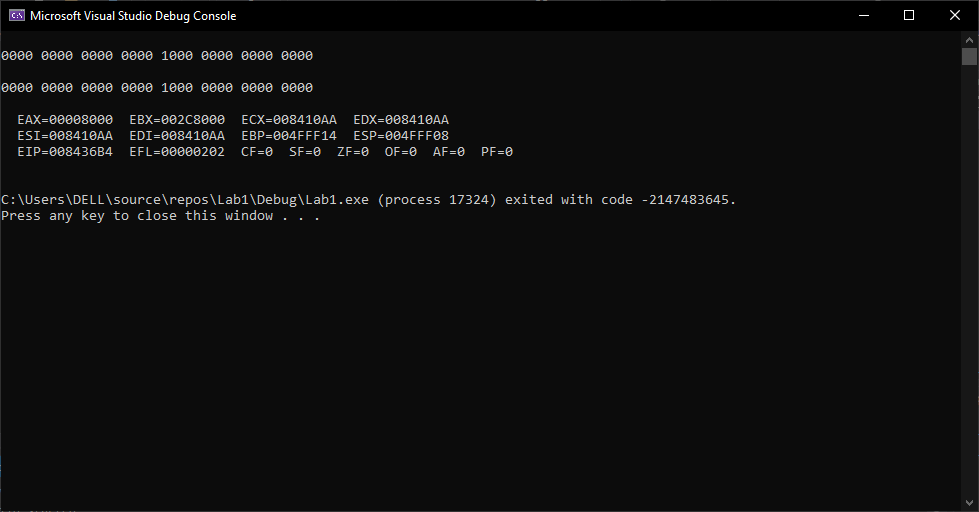
call writebin

call dumpregs

main ENDP

END main

Task4:



TITLE ReverseArray.asm

INCLUDE Irvine32.inc

.data

.code

main PROC

; without SHRD

mov eax, 0

mov ax, 0001b

mov bx, 0b

shl ax, 15

or bx, ax

mov eax, 0

mov ax, bx

call crlf

call writebin

call crlf

; With SHRD

mov eax, 0

mov ax, 0001b

mov bx, 0b

shrd bx, ax, 1

mov ax, bx

call crlf

call writebin

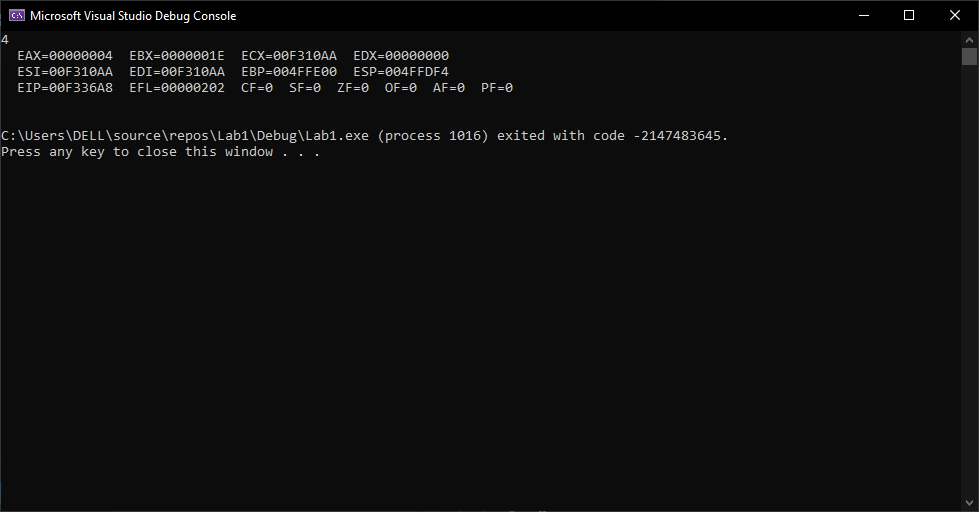
call crlf

call dumpregs

main ENDP

END main

Task5:



TITLE ReverseArray.asm

INCLUDE Irvine32.inc

.data

val1 sdword 60

val2 sdword 30

val3 sdword 15

tmp1 sdword 0

tmp2 sdword 0

.code

main PROC

;exp 1

mov edx, 0

mov eax, val2

mov ebx, val3

div ebx

mov tmp1, eax

;exp 2 k226007

mov edx, 0

mov eax, val1

mov ebx, val2

div ebx

mov tmp2, eax

;multiply

mov eax, tmp1

mul tmp2

mov val1, eax

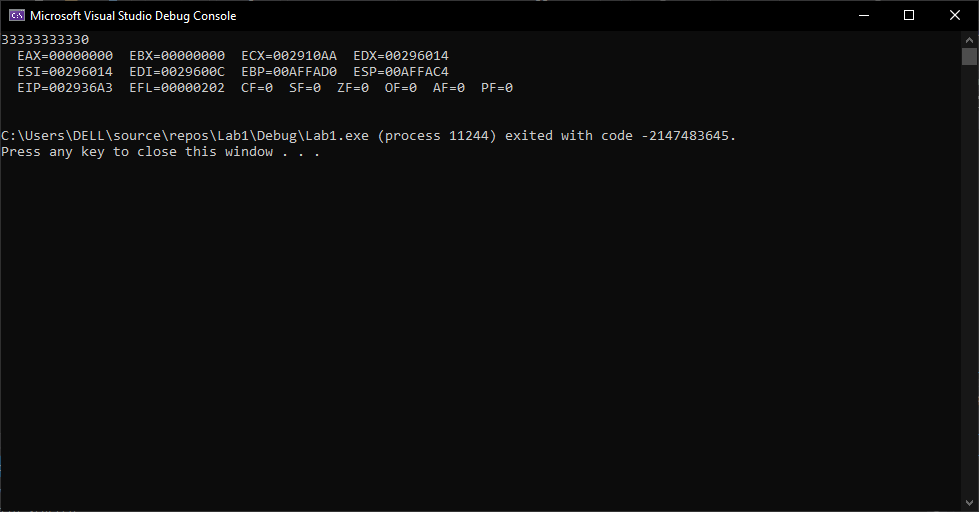
call writedec

call dumpregs

main ENDP

END main

Task6:



TITLE ReverseArray.asm

INCLUDE Irvine32.inc

.data

num1 qword 1111111111

num2 qword 2222222222

res qword 0

.code

main PROC

mov esi, offset num1

mov edi, offset num2

mov edx, offset res

; first 32

mov eax, [esi]

mov ebx, [edi]

add eax, ebx

mov [edx], eax

add esi, 4

add edi, 4

add edx, 4

;second 32

mov eax, [esi]

mov ebx, [edi]

add eax, ebx

mov [edx], eax

;display

mov esi, offset res

mov eax, [esi]

call writedec

add esi, 4

mov eax, [esi]

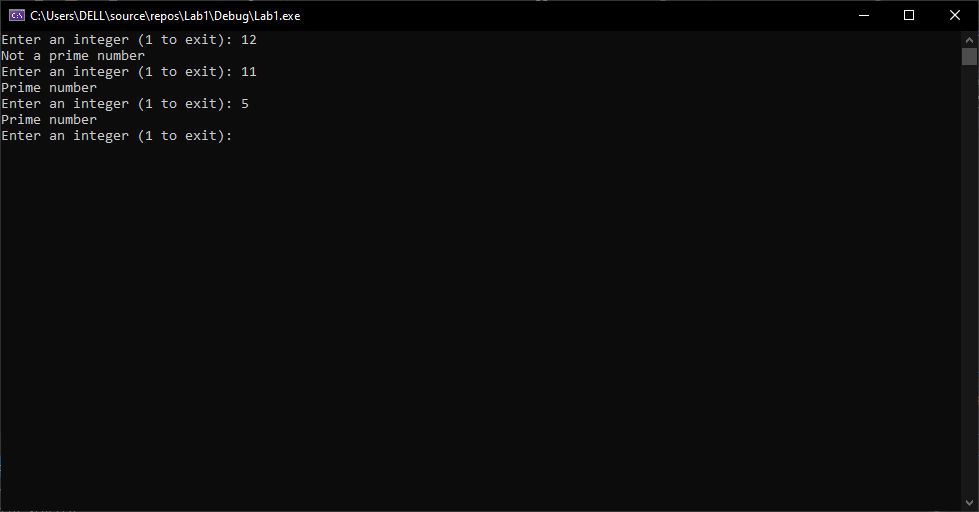
call writedec

call dumpregs

main ENDP

END main

Task7:



TITLE IsPrime.asm

INCLUDE Irvine32.inc

.data

prompt BYTE "Enter an integer (1 to exit): ", 0

primeMsg BYTE "Prime number", 0

notPrimeMsg BYTE "Not a prime number", 0

numberToDisplay SWORD 12345

anotherNumber SDWORD -987654321

.code

IsPrime PROC

cmp eax, 2

jl IsNotPrime

mov ecx, 2

CheckDivisor:

mov edx, 0

div ecx

cmp edx, 0

je IsNotPrime

inc ecx

cmp ecx, eax

jge IsPrimeDone

jmp CheckDivisor

IsNotPrime:

xor eax, eax

ret

IsPrimeDone:

ret

IsPrime ENDP

main PROC

InputLoop:

mov edx, OFFSET prompt

call WriteString

call ReadInt

cmp eax, 1

je ExitProgram

call IsPrime

jnz PrimeFound

mov edx, OFFSET notPrimeMsg

call WriteString

call Crlf

jmp InputLoop

PrimeFound:

mov edx, OFFSET primeMsg

call WriteString

call Crlf

jmp InputLoop

ExitProgram:

exit

main ENDP

DisplayNumbers PROC

movzx eax, numberToDisplay

mov edx, OFFSET prompt

call WriteString

call WriteInt

call Crlf

mov eax, anotherNumber

mov edx, OFFSET prompt

call WriteString

call WriteInt

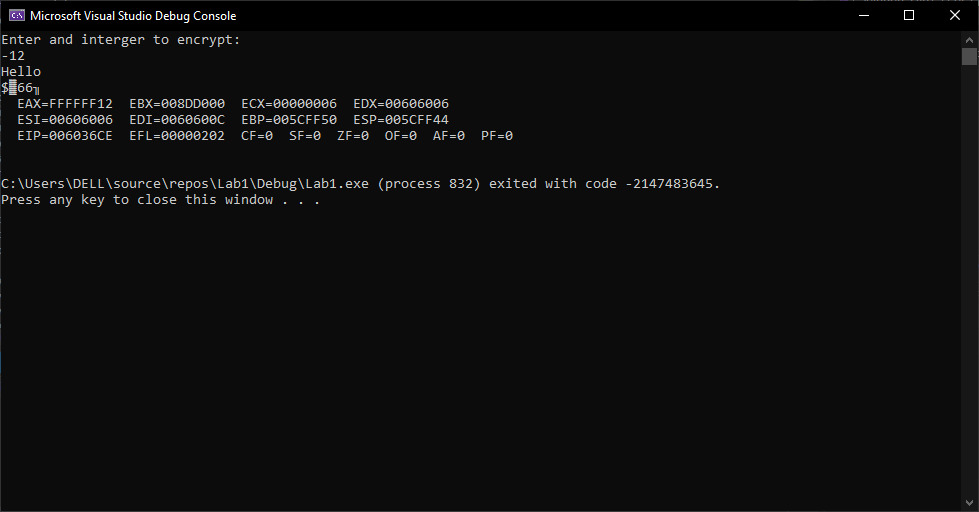
call Crlf

ret

DisplayNumbers ENDP

END main

Task8:



TITLE ReverseArray.asm

INCLUDE Irvine32.inc

.data

ostr byte "Hello",0

eyncstr byte 6 dup(?)

prompt byte "Enter and interger to encrypt: ",0

.code

main PROC

mov edx, offset prompt

call writestring

call crlf

call readint

test eax, 80h

js leftloop

mov esi, offset ostr

mov edi, offset eyncstr

mov ecx, 0

rightloop:

movzx edx, al

mov al, [esi]

ror al, 1

mov [edi], al

cmp ecx, edx

jg ended

inc esi

inc edi

inc ecx

jmp rightloop

leftloop:

movzx edx, al

neg edx

mov al, [esi]

rol al, 1

mov [edi], al

cmp ecx, edx

jg ended

inc esi

inc edi

inc ecx

jmp leftloop

ended:

;display

mov edx, offset ostr

call writestring

call crlf

mov edx, offset eyncstr

call writestring

call dumpregs

main ENDP

END main