**COAL LAB 08 (LAB TASKS)**

**TASK # 01:**

**CODE:**

Title Task1 (test.asm)

include irvine32.inc

.data

val1 dword 8h

val2 dword 6h

val3 dword 5h

message1 byte "The value of val1 after performing operations is : ",0

message2 byte "The value of ar4 after performing operations is : ",0

var1 dword -2h

var2 dword 5h

var3 dword 3h

ar4 dword ?

.code

main proc

mov edx, 0

mov eax, val2

div val3

mov ebx, eax

mov edx, 0

mov eax, val1

div val2

mov ecx, edx

mov edx, 0

mov eax, ebx

mul ecx

mov val1, eax

mov edx, offset message1

call writeString

call writeDec

call crlf

;ar4 = (var1 \* -5) / (-var2 % var3);

mov edx, 0

mov eax, var1

mov ebx, -5

imul ebx

mov ecx, eax

mov edx, 0

neg var2

mov eax, var2

mov ebx, var3

idiv ebx

mov ebx, edx

mov edx, 0

mov eax, ecx

div bx

mov edx, offset message2

call writeString

call writeDec

call crlf

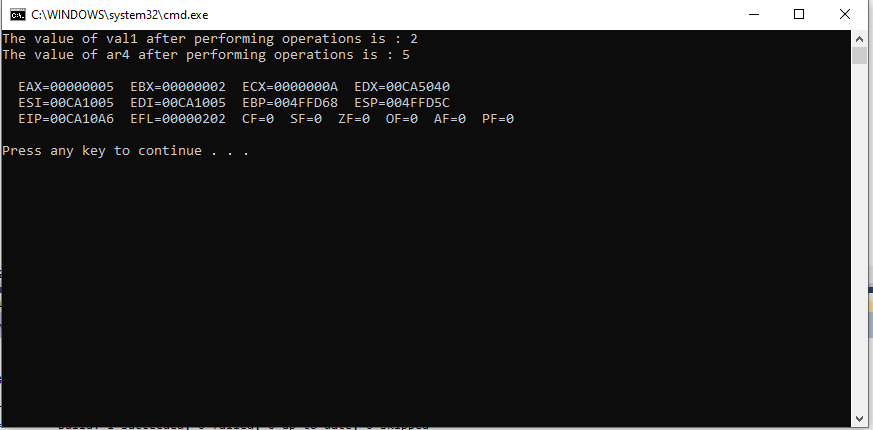
call dumpregs

exit

main endp

end main

**OUTPUT:**



**TASK # 02:**

**CODE:**

Title Task2 (test.asm)

include irvine32.inc

.data

X word 5

Y word 15

N word ?

message1 byte "The GCD of X and Y : ",0

.code

main proc

; first the time value of x : 5 and y : 15

call GCD

mov edx, offset message1

call writeString

mov eax, 0

mov ax, X

call writeDec

call crlf

; this time value of x : 8 and y : 12

mov eax, 0

mov ebx, 0

mov ax, 8

mov bx, 12

mov X, ax

mov Y, bx

call GCD

mov edx, offset message1

call writeString

mov eax, 0

mov ax, X

call writeDec

call crlf

; this time value of x : -7 and y : -35

mov eax, 0

mov ebx, 0

mov ax, -7

mov bx, -35

mov X, ax

mov Y, bx

call GCD

mov edx, offset message1

call writeString

mov eax, 0

mov ax, X

call writeDec

call crlf

call dumpregs

exit

main endp

GCD proc

mov eax, 0

mov ebx, 0

mov ax, X

mov bx, Y

cmp X, 0

JNL l1

neg X

l1: cmp Y, 0

JNL procedure

neg Y

procedure:

mov ax, X

mov bx, Y

mov edx, 0

div bx

mov N, dx

mov ax, Y

mov X, ax

mov Y, dx

mov bx, Y

cmp Y, 0

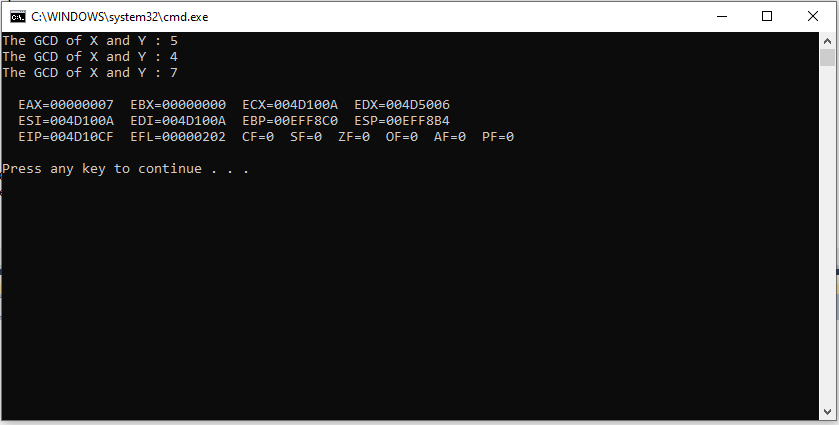
JG procedure

ret

GCD endp

end main

**OUTPUT:**



**TASK # 03:**

**CODE:**

Title Task2 (test.asm)

include irvine32.inc

.data

X Qword 01234567812345678h

Y Qword 04356892112398745h

result Qword ?

Message1 byte "The final answer after adding these 2 64 bits numbers is : ",0

.code

main proc

call Extended\_Add

mov edx, offset Message1

call WriteString

mov eax, dword ptr [result+4]

call writeHex

mov eax, dword ptr result

call writeHex

call crlf

call dumpregs

exit

main endp

Extended\_Add proc

mov eax, dword ptr X

mov edx, dword ptr [X+4]

add eax, dword ptr Y

adc edx, dword ptr [Y+4]

mov dword ptr result, eax

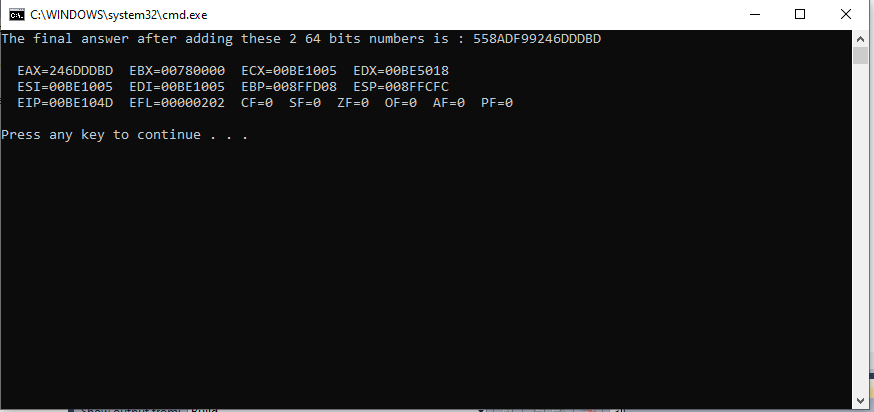
mov dword ptr [result+4], edx

ret

Extended\_Add endp

end main

**OUTPUT:**



**TASK # 04:**

**CODE:**

Title Task2 (test.asm)

include irvine32.inc

.data

key BYTE -2, 4, 1, 0, -3, 5, 2, -4, -4, 6

message1 byte "This plaintext message will be encrypted.",0

Original byte "Original Text : ",0

Encrypted byte "Encrypted Text: ",0

.code

main proc

mov edx, offset Original

call writeString

mov edx, offset message1

call writeString

call crlf

mov esi, 0

mov edi, 0

mov ecx, lengthof message1

function:

mov eax, ecx

cmp key[edi], 0

JNL l2

neg key[edi]

mov cl, key[edi]

rol message1[esi],cl

jmp nextt

l2: mov cl, key[edi]

ror message1[esi],cl

nextt:

inc esi

inc edi

cmp edi, 10

JNE continue

mov edi, 0

continue:

mov ecx, eax

loop function

mov edx, offset Encrypted

call writeString

mov edx, offset message1

call writeString

call crlf

call dumpregs

exit

main endp

end main

**OUTPUT:**

