**COAL LAB 12**

**Question 1**

**CPP code**

#include <stdio.h>

extern "C" void ThreeProd(), clear();

int main()

{

clear();

unsigned short num1, num2, num3;

unsigned long result;

printf("Enter the first number: \n");

scanf\_s("%hX", &num1);

printf("Enter the second number: \n");

scanf\_s("%hX", &num2);

printf("Enter the third number: \n");

scanf\_s("%hX", &num3);

\_asm

{

MOV AX, num1

MOV BX, num2

MOV CX, num3

call ThreeProd

MOV result, EAX

}

printf("\nThe Product of the three numbers is: %d.", result);

return 0;

}

**Assembly code**

.686

.MODEL FLAT, C

.STACK 2048

.DATA

.CODE

clear PROC

xor eax, eax

xor ebx, ebx

xor ecx, ecx

ret

clear ENDP

ThreeProd PROC

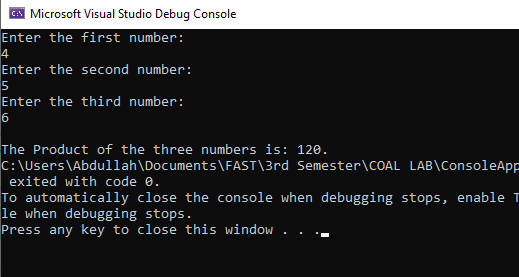
IMUL bx

IMUL cx

ret

ThreeProd ENDP

END

****

**Question 2**

**CPP code**

#include <stdio.h>

extern "C" void GCD(), clear();

int main()

{

clear();

unsigned long num1, num2, result;

printf("Enter number 1: ");

scanf\_s("%lu", &num1);

printf("Enter number 2: ");

scanf\_s("%lu", &num2);

\_asm

{

push num1

push num2

call GCD

MOV result, eax

}

printf("\nThe GCD of the two numbers is: %lu.", result);

return 0;

}

**Assembly code**

.686

.model flat, C

.stack 4096

.data

store DWORD ?

.code

clear PROC

xor eax, eax

xor ebx, ebx

ret

clear endp

GCD PROC

enter 0,0

MOV ebx, [ebp+8]

MOV eax, [ebp+12]

cmp eax, 0

JNE cond2

MOV eax, ebx

jmp endd

cond2:

cmp ebx, 0

JNE Process

jmp endd

process:

cmp eax, ebx

JE endd

cmp eax, ebx

JB rec2

SUB eax, ebx

push eax

push ebx

call GCD

jmp endd

rec2:

SUB ebx, eax

push eax

push ebx

call GCD

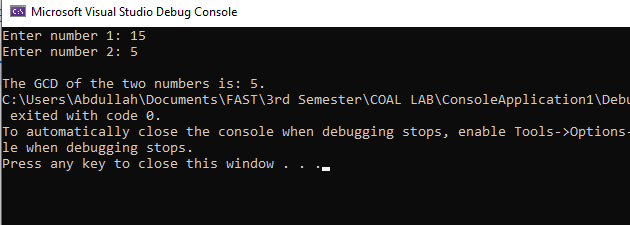
endd:

leave

ret 8

GCD endp

END

****