Task 1

.asm

Include Irvine32.inc

.model flat, C

.code

clear PROC

xor eax,eax

xor ebx,ebx

ret

clear ENDP

END

.cpp

#include <iostream>

using namespace std;

extern "C" void clear();

int main() {

clear();

unsigned long a, b, c, r;

cout << "Enter 3 numbers: ";

cin >> a >> b;

cin >> c;

\_asm {

mov eax, a

mov ebx, b

mov ecx,c

imul eax,ebx

imul eax,ecx

mov r,eax

}

cout << "Product : " << r;

return 0;

}

void clear() {

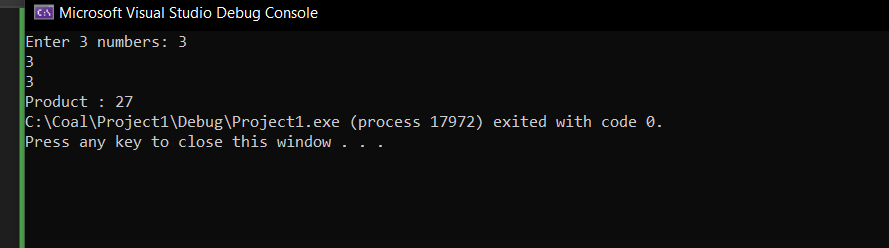
\_asm {

xor eax, eax

xor ebx, ebx

}

}



Task 2

.asm

INCLUDE Irvine32.inc

.model flat,C

.data

a DWORD ?

b DWORD ?

.code

PUBLIC GCD

GCD PROC

mov eax, a

mov ebx, b

GCD\_loop:

cmp eax, ebx

je GCD\_done

ja GCD\_a\_greater

jb GCD\_b\_greater

GCD\_a\_greater:

sub eax, ebx

jmp GCD\_loop

GCD\_b\_greater:

sub ebx, eax

jmp GCD\_loop

GCD\_done:

ret

GCD ENDP

main PROC

mov edx, OFFSET a

call ReadInt

mov edx, OFFSET b

call ReadInt

call GCD

mov edx, OFFSET a

mov eax, [edx]

; Print the result

call WriteInt

call Crlf

exit

main ENDP

END main

.c++

#include <iostream>

using namespace std;

extern "C" int GCD();

int main() {

int num1, num2;

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

int result = GCD(num1, num2); // Call the assembly function

cout << "GCD of " << num1 << " and " << num2 << " is: " << result << endl;

return 0;

}