**Name : Ibadullah Shaikh**

**Roll No : 19K-0259**

**COAL Lab 13 TASKS**

**Task 1:**

**ASM code:**

.686

.MODEL FLAT, C

.STACK 2048

.DATA

.CODE

GCD PROC

push ebp

mov ebp, esp

mov esi, [ebp+8]

mov edi, [ebp+12]

L1:

cmp esi, 1

jle one

cmp edi, 1

jle one

mov ecx, esi

L2:

mov edx, 0

mov eax, esi

div ecx

cmp edx, 0

jne done

mov edx, 0

mov eax, edi

div ecx

cmp edx, 0

jne done

mov eax, ecx

jmp one

done:

loop L2

jmp L1

one:

mov esp, ebp

pop ebp

ret

GCD ENDP

clear PROC

xor eax, eax

xor ebx, ebx

ret

clear ENDP

END

**CPP code:**

#include "stdafx.h"

extern "C" void clear();

extern "C" void GCD();

int main()

{

int one,two,res;

printf("Enter two Numbers:");

scanf\_s("%d %d", &one,&two);

\_asm

{

push one

push two

call GCD

add esp, 8

mov res, eax

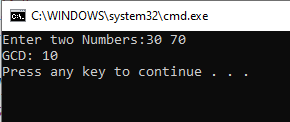
}

printf("GCD: %d\n", res);

return 0;

}

**Screenshot:**



**Task 2:**

**ASM Code:**

.686

.MODEL FLAT, C

.STACK 2048

.DATA

.CODE

ADDTHREE PROC

push ebp

mov ebp, esp

mov eax, 0

add eax, [ebp + 8]

add eax, [ebp + 12]

add eax, [ebp + 16]

mov esp, ebp

pop ebp

ret

ADDTHREE ENDP

clear PROC

xor eax, eax

xor ebx, ebx

ret

clear ENDP

END

**CPP Code:**

#include "stdafx.h"

extern "C" void clear();

extern "C" void ADDTHREE();

int main()

{

int one,two,three,res;

printf("Enter three Numbers:");

scanf\_s("%d %d %d", &one,&two,&three);

\_asm

{

push one

push two

push three

call ADDTHREE

add esp, 12

mov res, eax

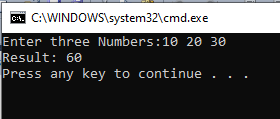
}

printf("Result: %d\n", res);

return 0;

}

**Screenshot:**



**Task 4:**

**ASM Code:**

.686

.MODEL FLAT, C

.STACK 2048

.DATA

.CODE

MinMax PROC

PUSH ebp

mov ebp, esp

mov ecx, [ebp + 8]

mov esi, [ebp + 12]

mov eax, DWORD PTR [esi]

dec ecx

add esi, 4

L1:

mov ebx, [esi]

cmp eax, ebx

jg L

mov eax, ebx

L:

add esi, 4

loop L1

PUSH eax

mov ecx, [ebp + 8]

mov esi, [ebp + 12]

mov eax, DWORD PTR [esi]

dec ecx

add esi, 4

L3:

mov ebx, [esi]

cmp eax, ebx

jl down

mov eax, ebx

down:

add esi, 4

loop L3

POP ebx

mov esp, ebp

pop ebp

ret

MinMax ENDP

clear PROC

xor eax, eax

xor ebx, ebx

ret

clear ENDP

END

**CPP Code:**

#include "stdafx.h"

extern "C" void clear();

extern "C" void MinMax();

int main()

{

int array[10]={1,2,3,4,5,6,7,8,9,10},n=5,min=0,max=0;

int \*address=&array[0];

\_asm

{

PUSH address

PUSH n

Call MinMax

add esp, 8

mov max, ebx

mov min, eax

}

printf("Array is: ");

for(int i=0;i<n;i++)

{

printf("%d ", array[i]);

}

printf("Minimun: %d\n Maximum: %d", min, max);

return 0;

}

**Screenshot:**

