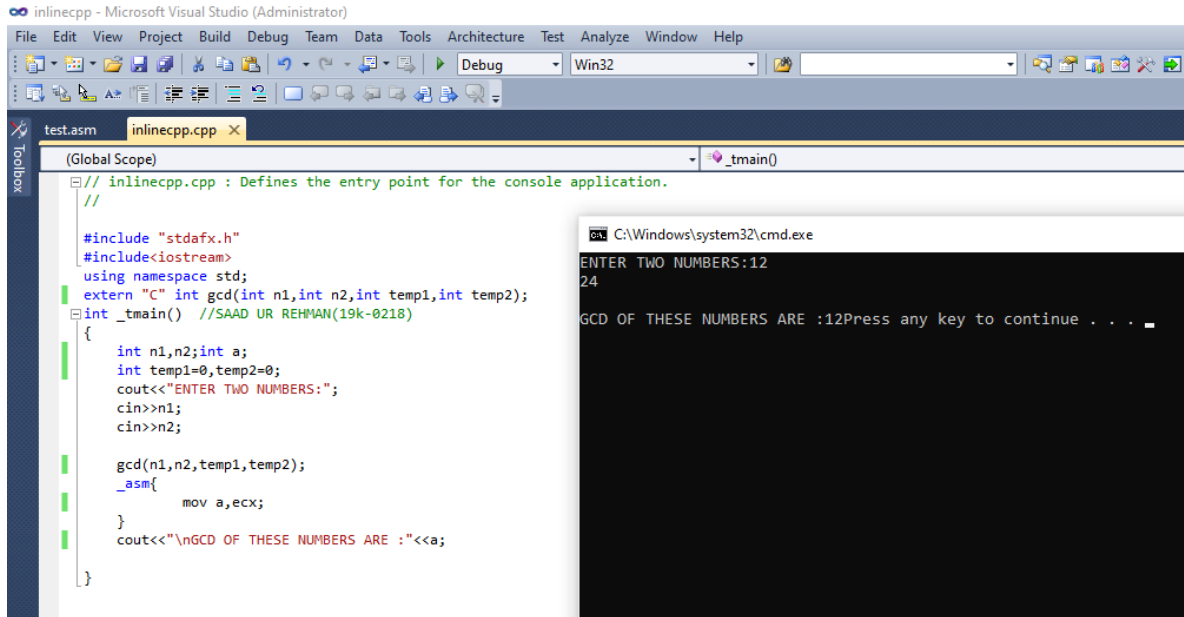


LAB 12 INLINE TASK

Q1:



The screenshot displays the Microsoft Visual Studio (Administrator) interface. The main window shows the source code for `inlinecpp.cpp`, which defines a recursive function `gcd` and its entry point `_tmain`. The code includes `stdafx.h` and `<iostream>`, uses the `std` namespace, and defines `gcd` as an external function. The `_tmain` function prompts the user to enter two numbers, reads them, and then calls `gcd` with an inline assembly block that moves the result into the `ecx` register. The output window on the right shows the program's execution, where the user has entered 12 and 24, and the program has output the GCD as 12.

```
// inlinecpp.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <iostream>
using namespace std;
extern "C" int gcd(int n1,int n2,int temp1,int temp2);
int _tmain() //SAAD UR REHMAN(19k-0218)
{
    int n1,n2;int a;
    int temp1=0,temp2=0;
    cout<<"ENTER TWO NUMBERS:";
    cin>>n1;
    cin>>n2;

    gcd(n1,n2,temp1,temp2);
    __asm{
        mov a,ecx;
    }
    cout<<"\nGCD OF THESE NUMBERS ARE : "<<a;
}
```

C:\Windows\system32\cmd.exe
ENTER TWO NUMBERS:12
24
GCD OF THESE NUMBERS ARE :12Press any key to continue . . .

inlinecpp - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

test.asm inlinecpp.cpp

```
.686
.MODEL FLAT, C
.STACK 2048
.CODE ;SAAD UR REHMAN(19k-0218)

gcd PROC,num1:DWORD,num2:DWORD,temp1:DWORD,temp2:DWORD
mov eax,num1
cmp eax,num2
mov ecx,num1
jb 11

11:
mov edx,0
mov eax,num1
mov ebx,num1
div ebx
mov temp1,edx

mov edx,0
mov ebx,0
mov eax,num2
mov ebx,num1
div ebx
mov temp2,edx

cmp temp1,0
je 12
jmp next
12:
cmp temp2,0
je 13
jmp next
```

C:\Windows\system32\cmd.exe

ENTER TWO NUMBERS:12
24

GCD OF THESE NUMBERS ARE :12Press any key to continue . . .

inlinecpp - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

test.asm inlinecpp.cpp

```
mov edx,0
mov ebx,0
mov eax,num2
mov ebx,num1
div ebx
mov temp2,edx

cmp temp1,0
je 12
jmp next
12:
cmp temp2,0
je 13
jmp next

next:
dec num1
dec num2

loop 11

13:
mov eax,ecx
ret
gcd ENDP
END
```

C:\Windows\system32\cmd.exe

ENTER TWO NUMBERS:12
24

GCD OF THESE NUMBERS ARE :12Press any key to continue . . .

Q2:

inlinecpp - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

test.asm inlinecpp.cpp

(Global Scope) _tmain()

```
// inlinecpp.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <iostream>
using namespace std;
extern "C" int threeadd(int n1,int n2,int n3);
int _tmain() //SAAD UR REHMAN(19k-0218)
{
    int n1,n2,n3;
    cout<<"ENTER THREE NUMBERS\n";
    cin>>n1;
    cin>>n2;
    cin>>n3;
    cout<<"\nSUM OF THESE THREE NUMBERS ARE : "<<threeadd(n1,n2,n3)<<"\n";
    return 0;
}
```

C:\Windows\system32\cmd.exe

ENTER THREE NUMBERS
10
2
4

SUM OF THESE THREE NUMBERS ARE :16
Press any key to continue . . .

inlinecpp - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

test.asm inlinecpp.cpp

(Global Scope) _tmain()

```
// inlinecpp.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <iostream>
using namespace std;
extern "C" int threeadd(int n1,int n2,int n3);
int _tmain() //SAAD UR REHMAN(19k-0218)
{
    int n1,n2,n3;
    cout<<"ENTER THREE NUMBERS\n";
    cin>>n1;
    cin>>n2;
    cin>>n3;
    cout<<"\nSUM OF THESE THREE NUMBERS ARE : "<<threeadd(n1,n2,n3)<<"\n";
    return 0;
}
```

C:\Windows\system32\cmd.exe

ENTER THREE NUMBERS
10
2
4

SUM OF THESE THREE NUMBERS ARE :16
Press any key to continue . . .

Q3:

inlinecpp - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

Debug Win32

test.asm inlinecpp.cpp

(Global Scope) _tmain()

```
// inlinecpp.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <iostream>
using namespace std;
extern "C" int binary_mult(int n1,int n2);
int _tmain() //SAAD UR REHMAN(19k-0218)
{
    int n1,n2;
    cout<<"ENTER TWO NUMBERS\n";
    cin>>n1;
    cin>>n2;
    cout<<"\nBINARY MULTIPLICATION OF THESE TWO NUMBERS ARE : "<<binary_mult(n1,n2)<<"\n";
    return 0;
}
```

C:\Windows\system32\cmd.exe

ENTER TWO NUMBERS
2
5

BINARY MULTIPLICATION OF THESE TWO NUMBERS ARE :10
Press any key to continue . . .

inlinecpp - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help

Debug Win32

test.asm inlinecpp.cpp

```
.686
.MODEL FLAT, C
.STACK 2048
.CODE ;SAAD UR REHMAN(19k-0218)

binary_mult PROC,num1:DWORD,num2:DWORD

    mov eax,num1
    mov ebx,num2
    mul ebx
    ret
binary_mult ENDP
END
```

C:\Windows\system32\cmd.exe

ENTER TWO NUMBERS
2
5

BINARY MULTIPLICATION OF THESE TWO NUMBERS ARE :10
Press any key to continue . . .

Q4:

The screenshot shows the Microsoft Visual Studio IDE with the file `inlinecpp.cpp` open. The code is a C++ program that uses `std::cout` to prompt the user to enter 10 numbers, reads them into an array, and then uses an assembly function `MinMaxArray` to find the minimum and maximum values. The program is running in the `Win32` architecture. The output window shows the execution results, including the input numbers and the calculated minimum and maximum values.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
extern "C" int MinMaxArray(int n[10]);
int _tmain() //SAAD UR REHMAN(19k-0218)
{
    int n[10];
    int min,max;
    cout<<"ENTER 10 NUMBERS";
    for(int i=0;i<10;i++)
    {
        cin>>n[i];
    }
    MinMaxArray(n);
    _asm{
        mov min,ebx;
        mov max,edx;
    }
    cout<<"\n MINIMUM VALUE:"<<min<<"\n MAXIMUM VALUE :"<<max;
    return 0;
}
```

Output:

```
C:\Windows\system32\cmd.exe
ENTER 10 NUMBERS1
23
2
4
5
65
7
23
7
8

MINIMUM VALUE:1
MAXIMUM VALUE :65Press any key to continue . . .
```

The screenshot shows the Microsoft Visual Studio interface with the 'inlinecpp' project open. The 'test.asm' file is active, displaying assembly code for finding minimum and maximum values in an array. A command prompt window is also open, showing the execution of the program with input '10' and output 'MINIMUM VALUE:1' and 'MAXIMUM VALUE :65'.

Assembly Code (test.asm):

```
.686
.MODEL FLAT, C
.STACK 2048
.CODE ;SAAD UR REHMAN(19k-0218)
MinMaxArray PROC,n:DWORD
;FOR FINDING MINIMUM
mov edi,n
mov eax,[edi]
mov ebx,eax
mov ecx,10
finding_min:
mov eax,[edi]
cmp eax,ebx
ja 13
mov ebx,eax
13:
add edi,4
loop finding_min
;FOR FINDING MAXIMUM
mov eax,0
mov edi,n
mov eax,[edi]
mov edx,eax
mov ecx,10
finding_max:
mov eax,[edi]
cmp eax,edx
jnb 14
mov edx,eax
14:
add edi,4
loop finding_max
ret
```

Command Prompt Output:

```
C:\Windows\system32\cmd.exe
ENTER 10 NUMBERS1
23
2
4
5
65
7
23
7
8

MINIMUM VALUE:1
MAXIMUM VALUE :65Press any key to continue . . .
```