**Question 01**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

array sdword 30,-40,20,65,80,45

message1 byte "Enter the first number j",0

message2 byte "Enter the second number k",0

message3 byte "The sum is ",0

j sdword 0

k sdword 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ecx, 2

loo1:

mov esi, offset array

mov edx, offset message1

call writestring

call crlf

call readint

mov j, eax

mov edx, offset message2

call writestring

call crlf

call readint

mov k, eax

mov eax, 0

mov ebx, ecx

call Calculate

mov ecx, ebx

loop loo1

call Dumpregs

exit

main ENDP

Calculate proc

mov ecx, 6

L1:

mov edx, j

cmp [esi], edx

jge L2

jmp l3

L2:

mov edx, k

cmp [esi], edx

jg l3

add eax,[esi]

l3:

add esi,4

loop L1

mov edx, offset message3

call writestring

call writedec

call crlf

ret

Calculate endp

END main

**Question 03**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

array dword 5,78,97,45,1,35,47,16,55

var dword 0

var1 dword 0

var2 dword 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ecx, 0

mov esi, 0

push offset array

push lengthof array

call BubbleSort

mov ecx, lengthof array

Loo:

mov eax, array[esi]

call writedec

call crlf

add esi,4

loop Loo

call Dumpregs

exit

main ENDP

BubbleSort Proc

push ebp

mov ebp, esp

mov esi, [ebp+12]

mov ecx, 0

mov ecx, [ebp+8]

mov edi,0

l1:

mov edx, ecx

mov ecx, lengthof array

sub ecx,1

mov edi,0

l2:

mov ebx, [esi+edi]

mov EAX, [esi+edi+4]

cmp ebx, eax

jg l3

add edi,4

loop l2

mov ecx, edx

loop l1

cmp ecx,0

jmp l4

l3:

mov var, ebx

mov ebx, eax

mov eax, var

mov [esi+edi], ebx

mov [esi+edi+4], eax

jmp l2

l4:

pop ebp

ret 8

BubbleSort Endp

END main

**Question 04**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

in\_message byte "Enter the number you want to calculate factorial of ", 0

out\_message byte "The value of the factorial of the given number is ", 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov edx, offset in\_message

call writestring

call crlf

call readint

mov ecx, eax

mov ebx,eax

l1:

dec ebx

cmp ebx,0

jz l2

mul ebx

loop l1

l2:

mov edx, offset out\_message

call writestring

call writedec

exit

main ENDP

END main

**Question 05**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

message1 byte "Enter a character to check its asci code: ",0

message2 byte "ASCI: ", 0

message3 byte "Number of 1's: ", 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ecx, 0

mov esi, 0

mov edx, offset message1

call writestring

call readchar

call writechar

call crlf

AND ah, bl

mov edx, offset message2

call writestring

call writebin

call crlf

mov ecx, 8

mov ebx, 0

L1:

SHR al,1

jnc een

add ebx,1

een:

loop L1

mov edx, offset message3

call writestring

mov eax, ebx

call writedec

call Dumpregs

exit

main ENDP

END main

**Question 06**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

CountMatches proto, ar1: ptr sdword, ar2: ptr sdword, s:dword

.data

message1 byte "The number of elements that are same in the first and the second array are ",0

message2 byte "The first array is: ", 0

message3 byte "The second array is ", 0

array1 sdword 1,3,5,7,9

array2 sdword 2,3,4,7,6

temp dword 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ecx, 0

mov esi, 0

mov edx, offset message2

call writestring

call crlf

mov ecx, lengthof array1

mov temp, ecx

mov esi, offset array1

L1:

mov eax, [esi]

call writeint

call crlf

add esi,4

loop L1

mov edx, offset message3

call writestring

call crlf

mov ecx, lengthof array2

mov esi, offset array2

L2:

mov eax, [esi]

call writeint

call crlf

add esi,4

loop L2

invoke CountMatches, addr array1, addr array2, lengthof array1

mov edx, offset message1

call writestring

call writedec

call crlf

call Dumpregs

exit

main ENDP

CountMatches proc, ar1: ptr sdword, ar2: ptr sdword, s:dword

mov eax,0

mov ecx, s

mov esi, 0

mov edi, 0

mov esi, offset array1

mov edi, offset array2

l1:

mov ebx, [esi]

mov edx, [edi]

cmp ebx, edx

jne eed

add eax,1

eed:

add esi,4

add edi,4

loop l1

ret 12

CountMatches endp

END main

**Question 07**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

Extended\_sub proto, ar1: sdword, ar2: sdword

.data

message1 byte "Subtracting 45 and 35",0

message2 byte "Subtracting 6560 and 250", 0

message3 byte "Subtracting 789 and 98", 0

message4 byte "The answer is ", 0

array1 sdword 1,3,5,7,9

array2 sdword 2,3,4,7,6

temp dword 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ecx, 0

mov esi, 0

mov edx, offset message1

call writestring

call crlf

invoke Extended\_sub, 45,35

mov edx, offset message2

call writestring

call crlf

invoke Extended\_sub, 6560,250

mov edx, offset message3

call writestring

call crlf

invoke Extended\_sub, 789,98

call Dumpregs

exit

main ENDP

Extended\_sub proc, ar1: sdword, ar2: sdword

mov eax, ar1

mov ebx, ar2

sub eax, ebx

mov edx, offset message4

call writestring

call writedec

call crlf

ret 8

Extended\_sub endp

END main

**Question 8**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

Extended\_add proto, ar1: sdword, ar2: sdword

.data

message1 byte "Adding 45 and 35",0

message2 byte "Adding 6560 and 250", 0

message3 byte "Adding 789 and 98", 0

message4 byte "The answer is ", 0

array1 sdword 1,3,5,7,9

array2 sdword 2,3,4,7,6

temp dword 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ecx, 0

mov esi, 0

mov edx, offset message1

call writestring

call crlf

invoke Extended\_add, 45,35

mov edx, offset message2

call writestring

call crlf

invoke Extended\_add, 6560,250

mov edx, offset message3

call writestring

call crlf

invoke Extended\_add, 789,98

call Dumpregs

exit

main ENDP

Extended\_add proc, ar1: sdword, ar2: sdword

mov eax, ar1

mov ebx, ar2

add eax, ebx

mov edx, offset message4

call writestring

call writedec

call crlf

ret 8

Extended\_add endp

END main

**Question 9**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

GCD\_Cal proto, ar1: sdword, ar2: sdword

.data

message1 byte "Case 01: GCD (5,20)",0

message2 byte "Case 01: GCD (24,18)", 0

message3 byte "Case 01: GCD (432,226)", 0

message4 byte "The GCD is ", 0

array1 sdword 1,3,5,7,9

array2 sdword 2,3,4,7,6

temp dword 0

.code

main PROC

mov eax, 0

mov ebx, 0

mov ecx, 0

mov edx, 0

mov ecx, 0

mov esi, 0

mov edx, offset message1

call writestring

call crlf

mov eax,5

mov ebx,20

invoke GCD\_Cal, eax, ebx

mov edx, offset message4

call writestring

call writedec

call crlf

mov edx, offset message2

call writestring

call crlf

mov eax,24

mov ebx,18

invoke GCD\_Cal, eax, ebx

mov edx, offset message4

call writestring

call writedec

call crlf

call Dumpregs

exit

main ENDP

GCD\_Cal proc, ar1: sdword, ar2: sdword

mov eax, ar1

mov ebx, ar2

cmp eax,0

jne loo2

ret 8

loo2:

cmp ebx,0

jne loo3

ret 8

loo3:

cmp eax, ebx

jne loo4

ret 8

loo4:

cmp eax, ebx

jle loo5

sub eax, ebx

invoke GCD\_Cal, eax, ebx

loo5:

sub ebx,eax

invoke GCD\_Cal, eax, ebx

ret 8

GCD\_Cal endp

END main