Q1)

INCLUDE irvine32.inc

.data

array WORD 10 DUP(?)

array2 WORD 1,2,3,4,5,6,7,8,9,10

.code

main proc

mov ecx,10

L1:

push array2[ecx]

LOOP L1

mov ecx,10

mov esi,0

L2:

pop array[esi]

add esi,2

LOOP L2

mov ecx,10

mov eax,0

mov esi,0

L3:

movsx eax,array[esi]

add esi,2

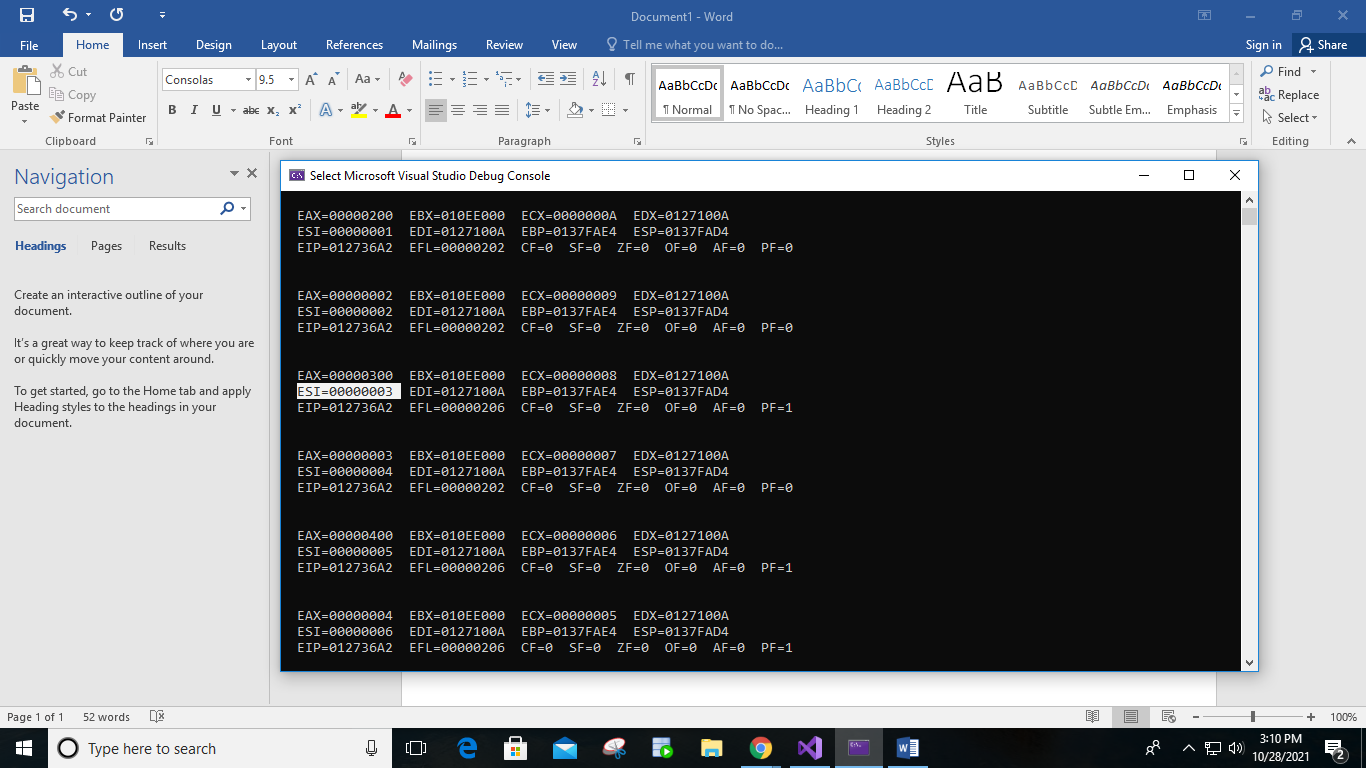
call Dumpregs

Loop L3

exit

main endp

end main



Q2)

INCLUDE irvine32.inc

.data

a WORD 1h

b WORD 2h

f WORD 3h

.code

main proc

push f

push b

push a

mov eax,0

pop eax

mov ebx,0

pop ebx

mov ecx,0

pop ecx

add eax,ebx

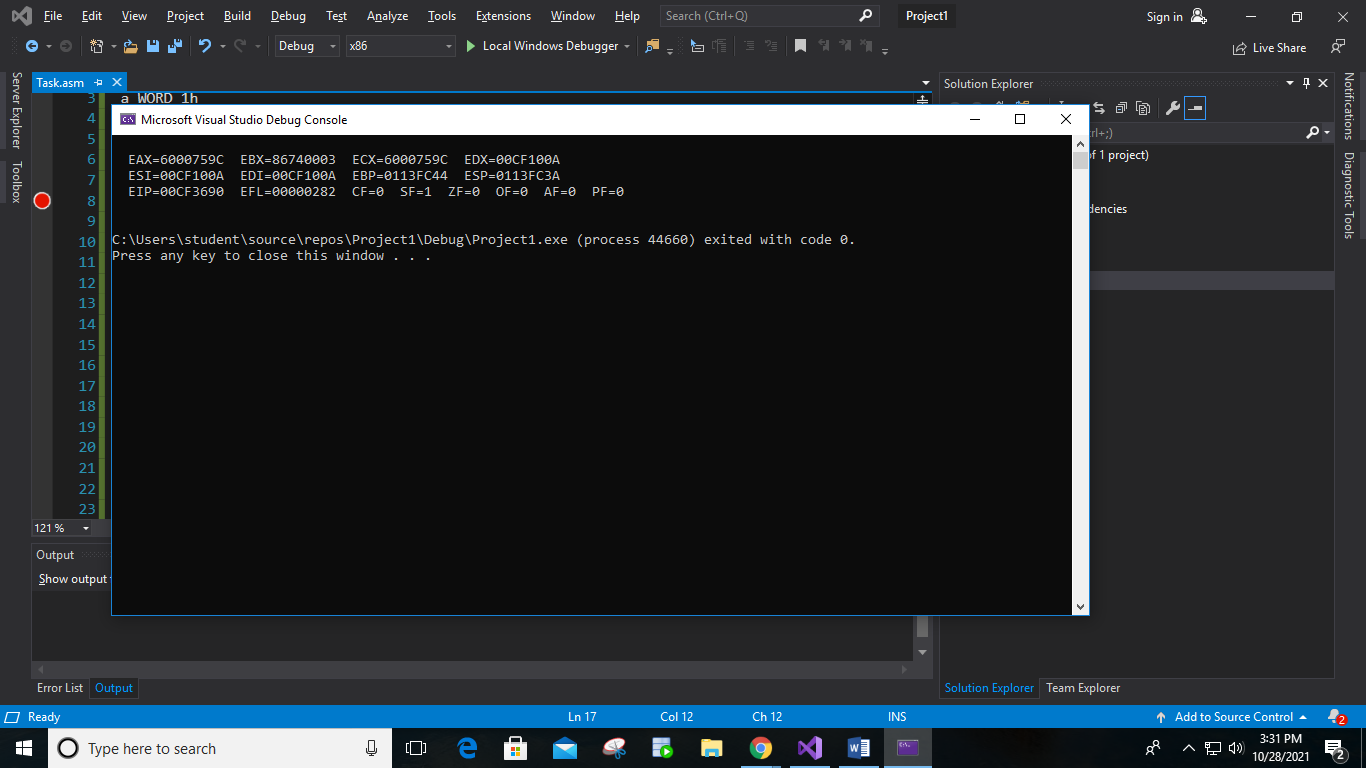
mov eax,ecx

call dumpregs

exit

main endp

end main



Q3

INCLUDE Irvine32.inc

.data

Array1 dword 1,2,3,4,5

Array2 dword 6,7,8,9,10

.code

main proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

mov esi,0

call AddArray1

exit

main endp

AddArray2 proc

mov esi,0

mov eax,0

mov ecx,5

L2:

add ebx, [Array2+esi]

add esi,4

Loop L2

mov eax, ebx

call AddArrays

ret

AddArray2 endp

AddArray1 proc

mov ecx,5

L1:

add eax, [Array1+esi]

add esi,4

Loop L1

mov edx, eax

call AddArray2

ret

AddArray1 endp

AddArrays proc

mov ecx, edx

add ecx, ebx

mov eax, ecx

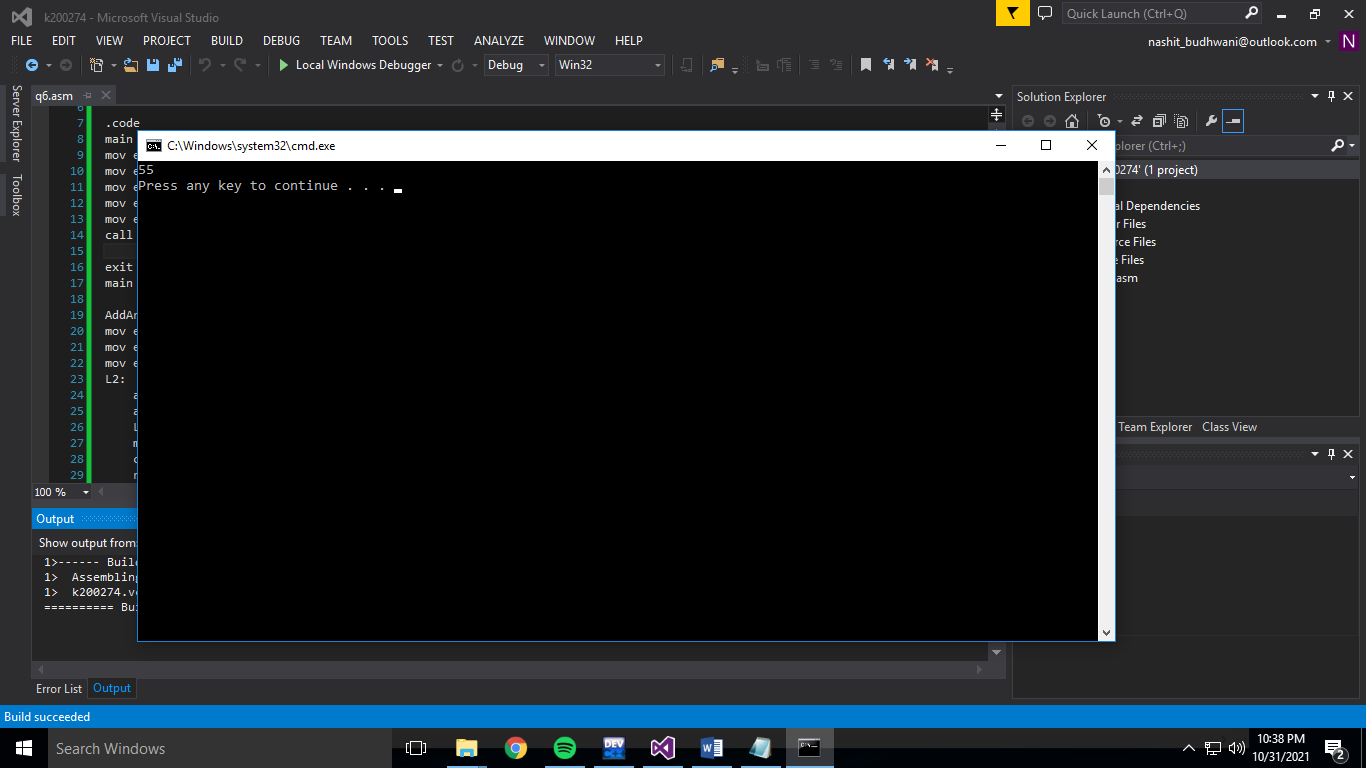
call Writedec

call crlf

ret

AddArrays endp

end main



Q4)

INCLUDE Irvine32.inc

.data

star dword '\*'

var1 BYTE 1

var2 BYTE 5

var3 BYTE 1

.code

main PROC

mov ecx,5

L1:

mov dl,var2

mov dh,var1

call Gotoxy

mov ebx,ecx

movzx ecx,var3

L2:

mov edx,OFFSET star

call WriteString

LOOP L2

mov ecx,ebx

inc var3

inc var1

dec var2

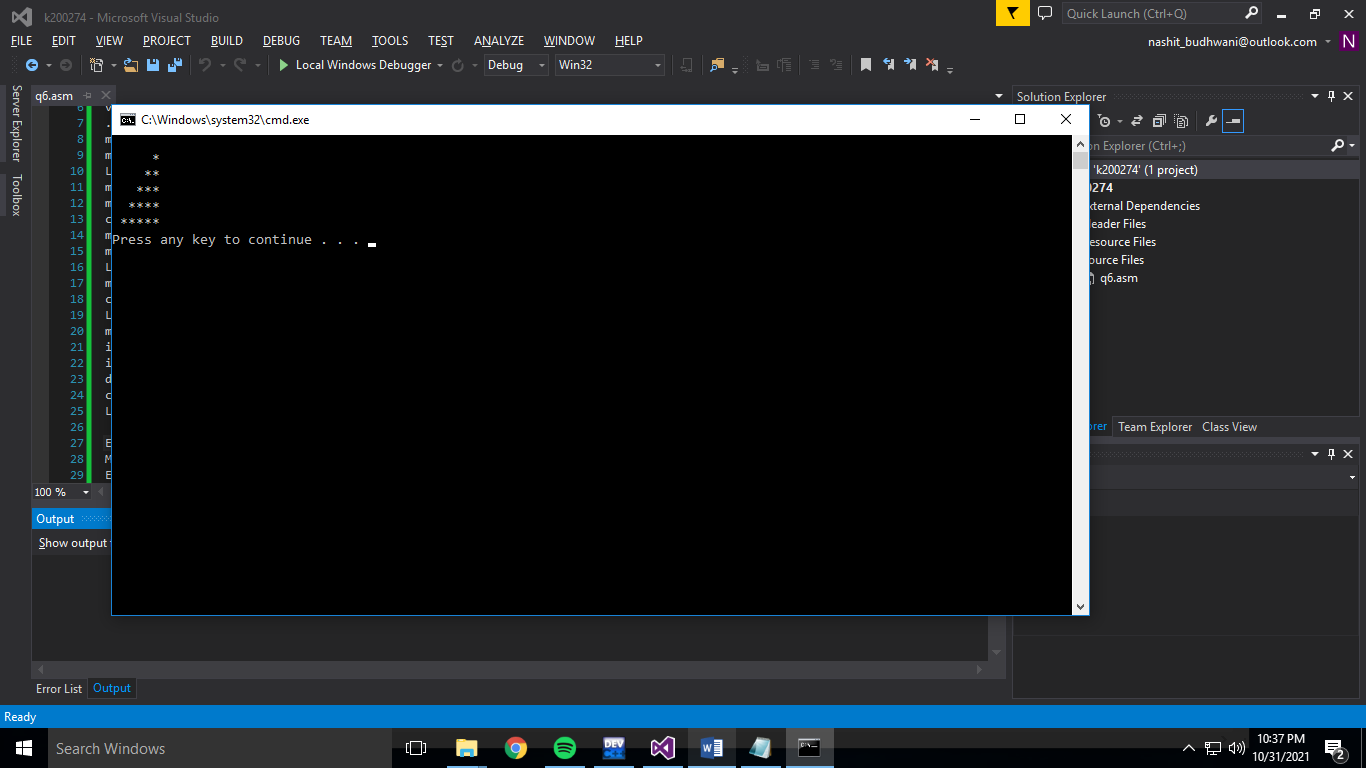
call crlf

Loop L1

EXIT

MAIN ENDP

END main



Q5)

INCLUDE Irvine32.inc

.data

star dword 'A'

var1 BYTE 1

var2 BYTE 5

var3 BYTE 1

.code

main PROC

mov ecx,5

L1:

mov dl,var2

mov dh,var1

call Gotoxy

mov ebx,ecx

movzx ecx,var3

L2:

mov edx,OFFSET star

call WriteString

inc star

LOOP L2

mov ecx,ebx

inc var3

inc var1

dec var2

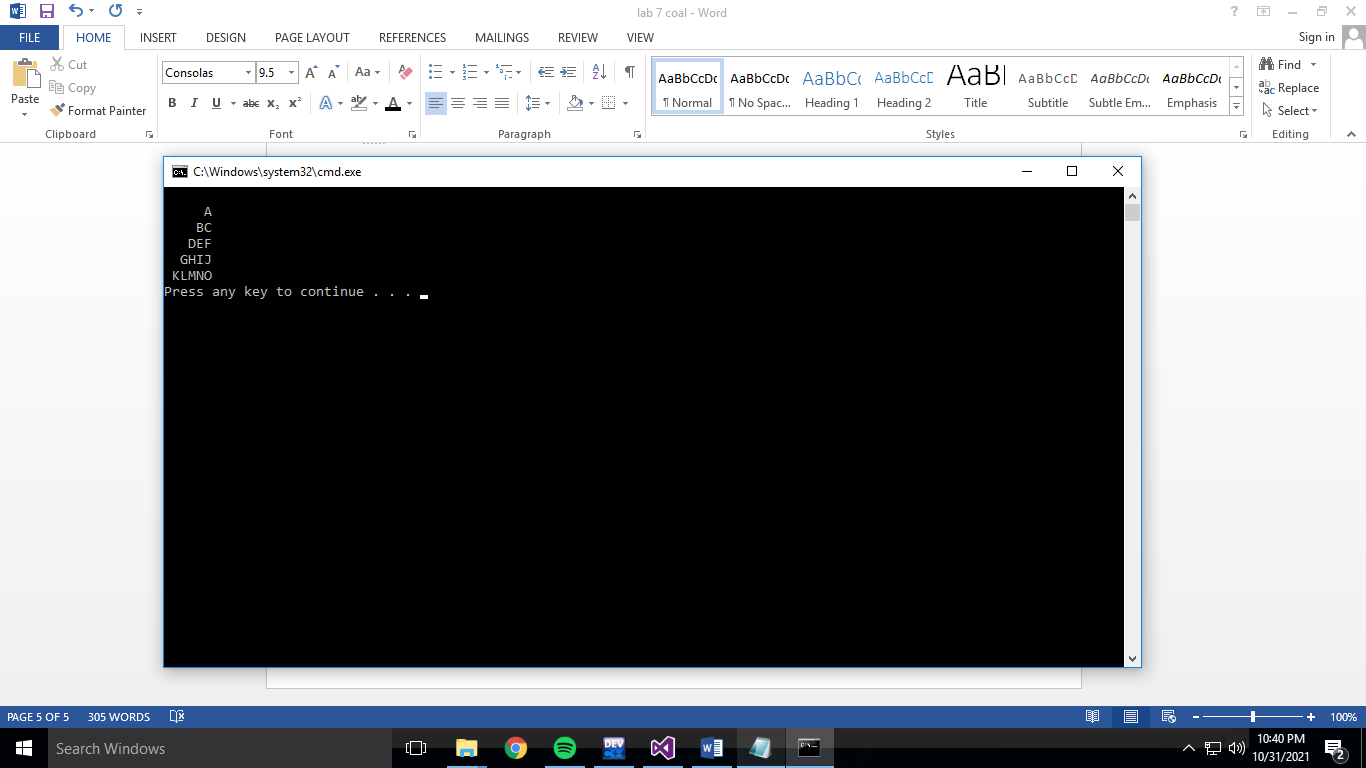
call crlf

Loop L1

EXIT

MAIN ENDP

END main



Q6)

INCLUDE Irvine32.inc

.code

main proc

Call func

exit

main endp

func Proc

mov eax, 0

mov ebx, 0

call readint

mov ecx, eax

mov ebp, eax

Loop1:

push eax

sub eax, 1

loop Loop1

mov ecx, ebp

Loop2:

pop edx

add ebx, edx

loop Loop2

mov eax, ebx

call writedec

call crlf

ret

func Endp

end main

