**COAL LAB 8**

**21k-4834**

**TASK 1**

include irvine32.inc

.data

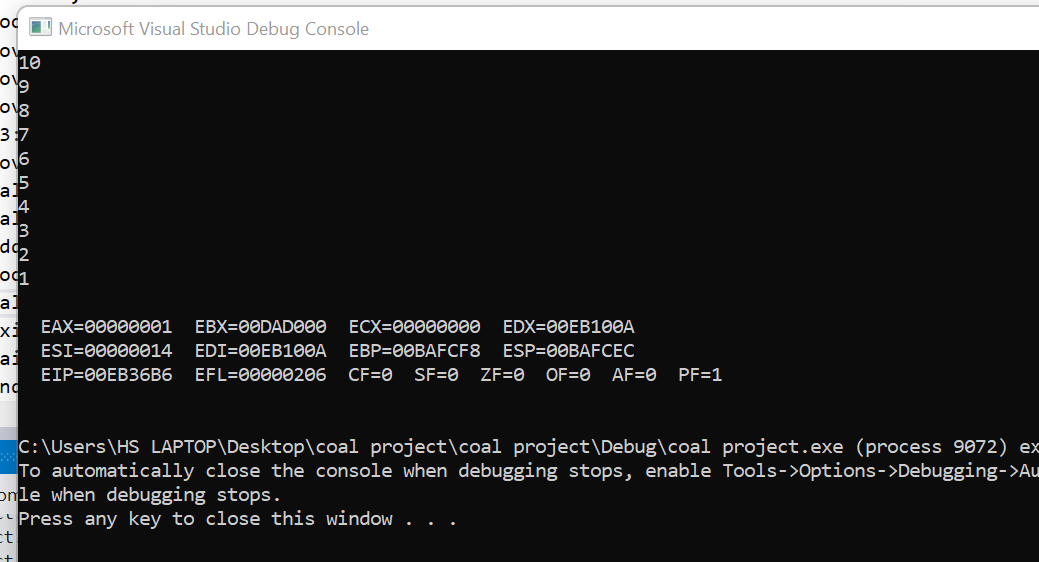
arr1 word 1,2,3,4,5,6,7,8,9,10

arr2 word 10 DUP(?)

.code

main proc

mov ecx,10



mov esi,0

L1:

push arr1[esi]

add esi,2

loop L1

mov esi,0

mov ecx,10

L2:

pop arr2[esi]

add esi,2

loop L2

mov ecx,10

mov esi,0

mov eax,0

L3:

mov ax,arr2[esi]

call writedec

call crlf

add esi,2

loop L3

call dumpregs

exit

main endp

end main

**TASK 4**

Include Irvine32.inc

.data

prompt1 byte "enter the number of columns",0

col DWORD ?

.code

Main PROC

mov edx, offset prompt1

call writestring

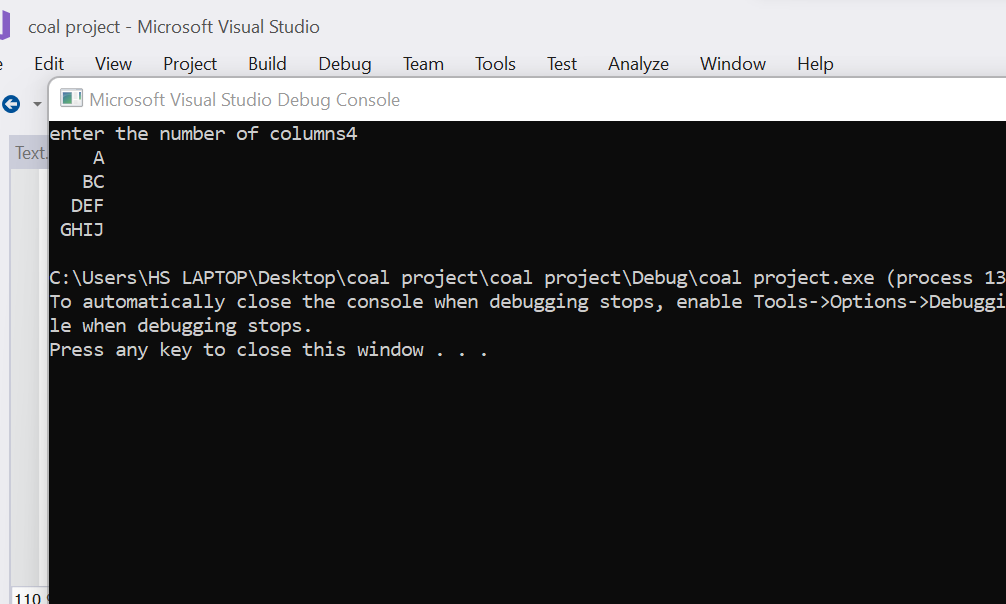
call readint

mov ecx,eax

call pattern

exit

main ENDP



pattern PROC

mov esi, 'A'

mov ebx,1

L1:

push ecx

L3:

mov eax,' '

call writechar

loop L3

mov ecx,ebx

L2:

mov eax,esi

call writechar

inc esi

loop L2

call crlf

pop ecx

inc ebx

Loop L1

ret

pattern ENDP

END Main

**TASK 5**

Include Irvine32.inc

.data

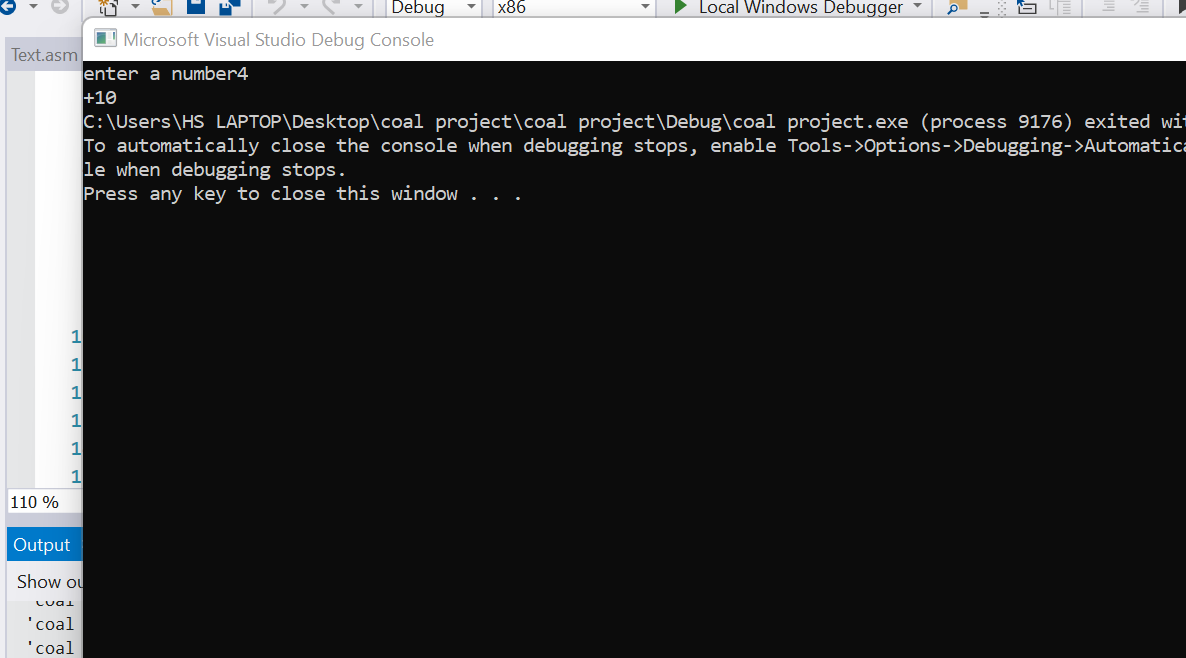
num DWORD ?

prompt1 byte "enter a number",0

.code

main PROC

mov edx, offset prompt1



call writestring

call readint

mov num,eax

call adding

exit

main ENDP

adding PROC

mov ebx,0

inc num

mov ecx,num

mov esi,0

mov eax,0

L1:

add eax,ebx

inc ebx

Loop L1

call writeint

ret

adding ENDP

end main

**TASK 2**

include irvine32.inc

.data

arr1 dword 1,2,3,4,5,213,23,12

arr2 dword 8,76,4,5,32,4

result dword ?

.code

main proc

call sumarrays

mov eax, result

call writeint

exit

main endp

sumarrays proc

call sumarr1

call sumarr2

add eax, ebx

mov result, eax

ret

sumarrays endp

sumarr1 proc

mov esi, 0

mov ecx, lengthof arr1

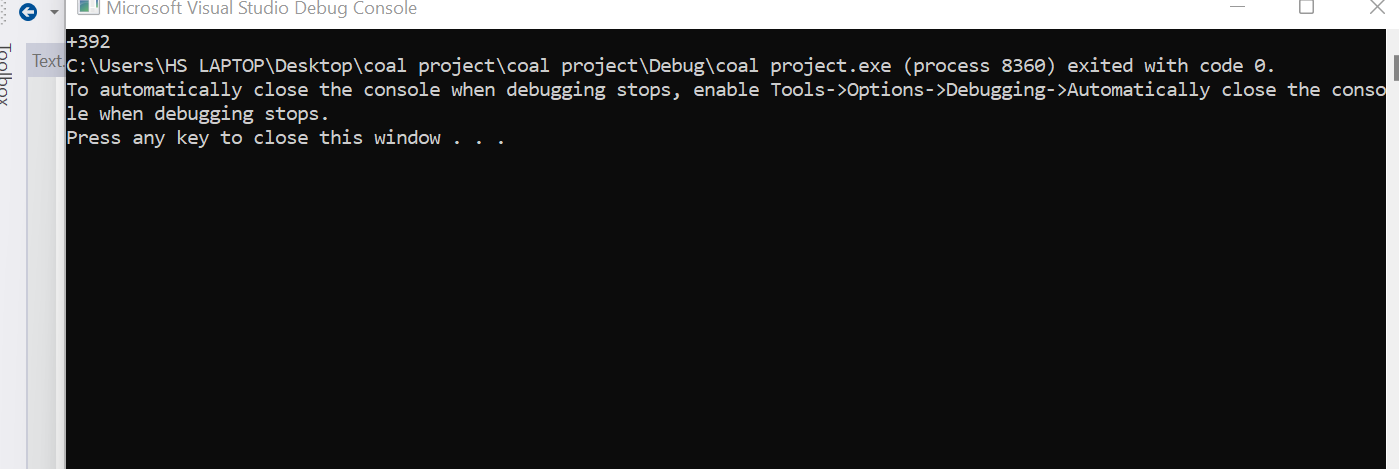
mov eax, 0

l1:

add eax, arr1[esi]

add esi, 4

loop l1



ret

sumarr1 endp

sumarr2 proc

mov esi, 0

mov ecx, lengthof arr2

mov ebx, 0

l2:

add ebx, arr2[esi]

add esi, 4

loop l2

ret

sumarr2 endp

end main

**TASK 3**

include irvine32.inc

.data

var dword 5

.code

main proc

call pattern

exit

main endp

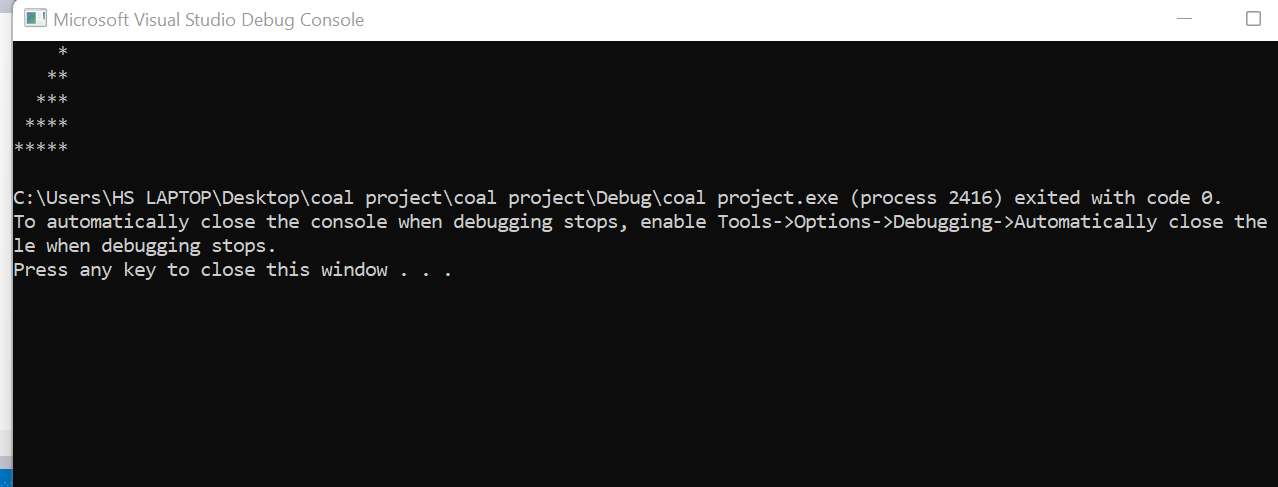
pattern proc

mov ecx, var

mov ebx, ecx

mov edx, 1

l1:



mov eax, ' '

push ecx

dec ebx

mov ecx, ebx

cmp ecx, 0

jle l3

l2:

call writechar

loop l2

l3:

mov eax, "\*"

mov ecx, edx

l4:

call writechar

loop l4

inc edx

call crlf

pop ecx

loop l1

ret

pattern endp

end main