**LAB 08**

**TASK 1**: *Take an array of 10 numbers move word-type of data into another empty array using stack push and pop technique*.

INCLUDE Irvine32.inc

.data

arr1 WORD 23,5,7,84,3,3,5,6,7,8

arr2 WORD 10 DUP(0)

str1 byte "the arrayu copied is:"

.code

main PROC

mov esi,0

mov ecx,LENGTHOF arr1

l1:

push arr1[esi]

add esi,2

loop l1

mov esi,9\*type arr2

mov ecx,10

l2:

pop arr2[esi]

sub esi,2

loop l2

mov ecx,10

mov esi,0

mov eax,0

mov edx,offset str1

call writestring

call crlf

l3:

mov ax,arr2[esi]

call writedec

call crlf

add esi,2

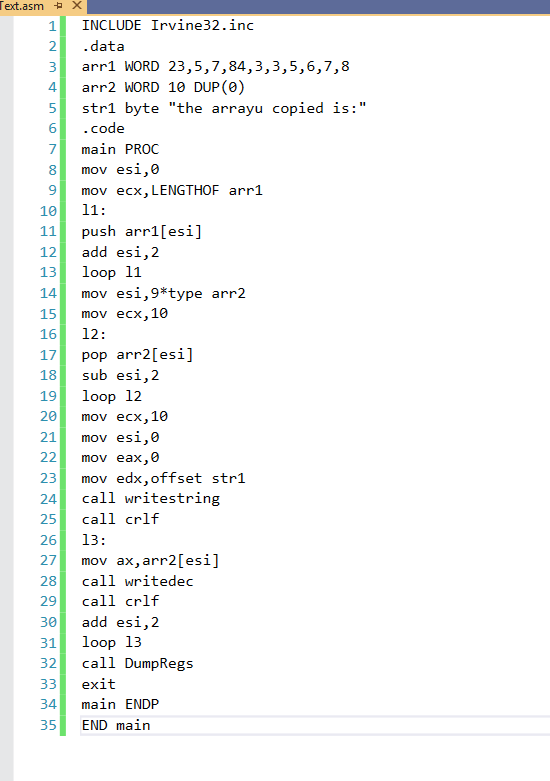
loop l3

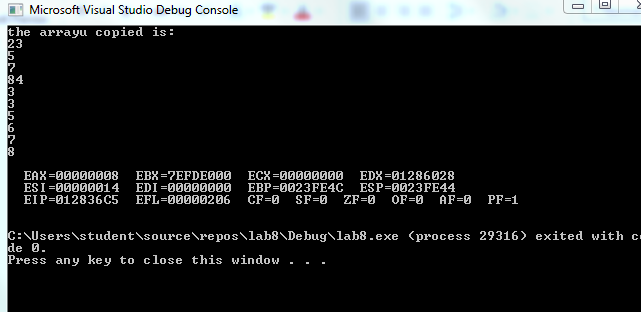
call DumpRegs

exit

main ENDP

END main





**TASK 2** : *Write a program having nested procedures are used to calculate the total sum of 2 arrays(each array having 5 elements). The sum of 1-array in 1st procedure and in 2nd procedure have sum of 2-array. And the 3rd procedure added the result of both.*

INCLUDE Irvine32.inc

.data

arr1 word 1,1,1,1,1

arr2 word 2,2,2,2,2

sum1 word ?

sum2 word ?

sum3 word ?

m1 byte "sum of array1 = ",0

m2 byte "sum of array2 = ",0

m3 byte "total sum:",0

.code

main PROC

call sumofarray1

mov edx,offset m1

call writestring

mov eax,0

mov ax,sum1

call writedec

call crlf

mov edx,offset m2

call writestring

mov ax,sum2

call writedec

call crlf

mov edx,offset m3

call writestring

mov ax,sum3

call writedec

call crlf

call DumpRegs

exit

main endp

sumofarray1 proc

mov ecx,5

mov esi,0

mov ax,0

mov bx,0

mov dx,0

l1:

mov ax,sum1

add ax,arr1[esi]

mov sum1,ax

call sumofarray2

add esi,2

loop l1

call total

ret

sumofarray1 endp

sumofarray2 proc

mov bx,sum2

add bx,arr2[esi]

mov sum2,bx

ret

sumofarray2 endp

total proc

mov dx,sum1

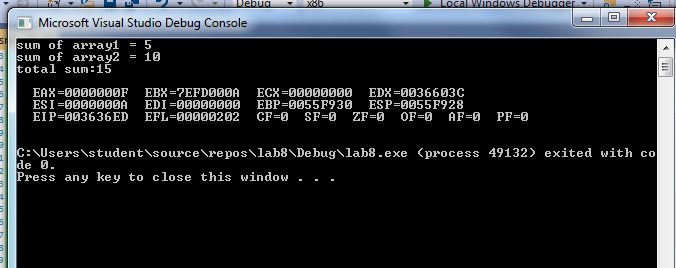
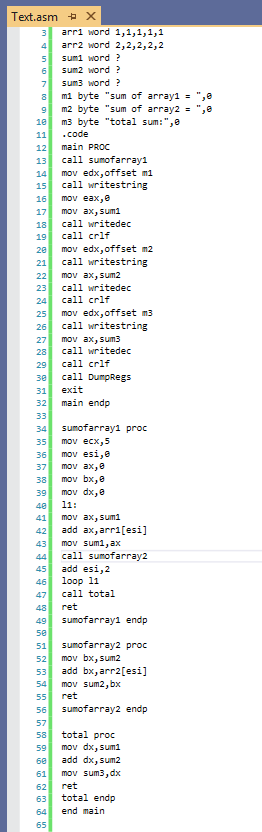
add dx,sum2

mov sum3,dx

ret

total endp

end main



**TASK 3:** *Print the following pattern using a function call in which number of columns is pass through a variable.*

INCLUDE irvine32.inc

.data

m1 byte "Enter number of rows: ",0

m2 byte "Sum of array 2: ",0

value word ?

v dword 0

v1 byte "\*",0

s byte " ",0

.code

main PROC

mov edx,offset m1

call writestring

mov eax,0

call readdec

mov value,ax

call prints

call DumpRegs

exit

main endp

prints PROC

mov ecx,0

mov cx,value

mov ebx,1

a1:

push ecx

b1:

mov edx,offset s

call writestring

loop b1

mov ecx,ebx

c1:

mov edx,offset v1

call writestring

loop c1

inc ebx

pop ecx

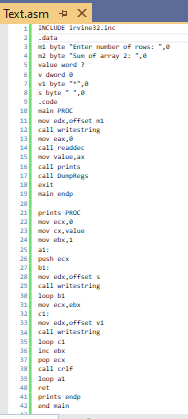
call crlf

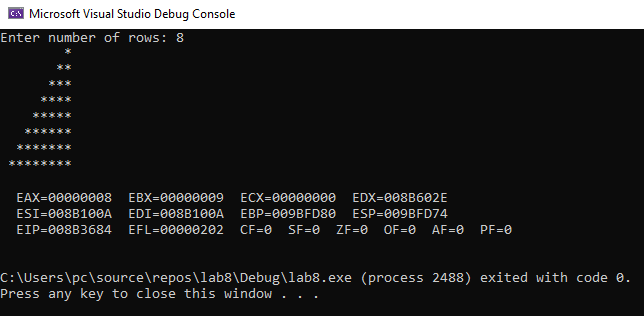
loop a1

ret

prints endp

end main





**TASK 4:** *Print the following pattern using a function call in which number of columns is pass through a variable.*

INCLUDE irvine32.inc

.data

m1 byte "Enter number of rows: ",0

m2 byte "Sum of array 2: ",0

value word ?

v dword 0

v1 byte "A",0

s byte " ",0

.code

main PROC

mov edx,offset m1

call writestring

mov eax,0

call readdec

mov value,ax

call prints

call DumpRegs

exit

main endp

prints PROC

mov ecx,0

mov cx,value

mov ebx,1

a1:

push ecx

b1:

mov edx,offset s

call writestring

loop b1

mov ecx,ebx

c1:

mov edx,offset v1

call writestring

add v1,1

loop c1

inc ebx

pop ecx

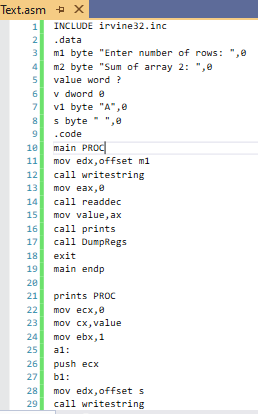
call crlf

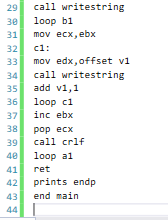
loop a1

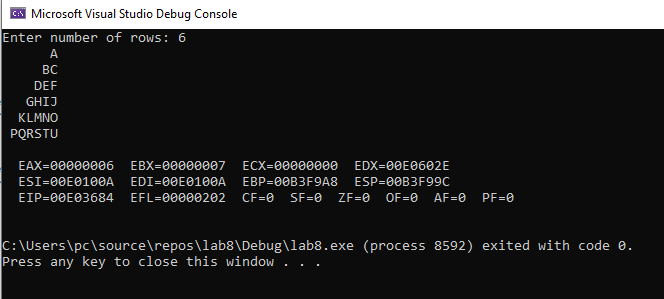
ret

prints endp

end main







**TASK 5:** *Write a function that asks the user for a number n and prints the sum of the numbers 1 to n.*

INCLUDE irvine32.inc

.data

m1 byte "Enter number : ",0

m2 byte "Sum from 1 till that number: ",0

value word ?

total word 0

.code

main PROC

mov edx,offset m1

call writestring

mov eax,0

call readdec

mov value,ax

call addtosum

mov edx,offset m2

call writestring

call writedec

call DumpRegs

exit

main endp

addtosum PROC

mov ecx,0

mov cx,value

mov ax,0

l1:

add ax,cx

loop l1

ret

addtosum endp

end main

