QUESTION 1:

include irvine32.inc

.data

n word 20

.code

main PROC

mov ax, 0

mov cx, n

l1:

add ax,cx

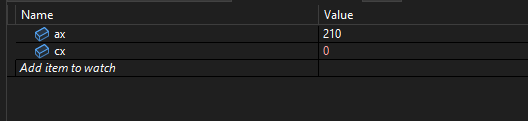
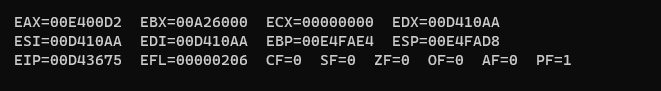
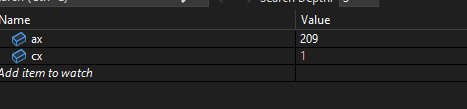
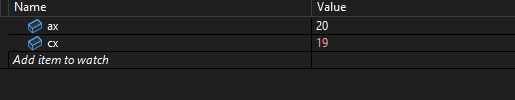
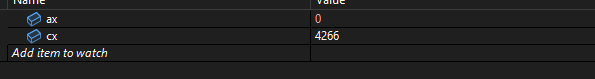
loop l1

call DumpRegs

exit

main ENDP

END main



QUESTION 3:

INCLUDE Irvine32.inc

.data

result DD ?

.code

main PROC

mov eax, 1

mov ecx, 10

L1:

mul ecx ;

loop L1 ; repeat until 0

mov result, eax ; store the result

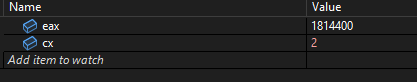
call DumpRegs

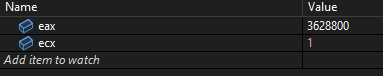
exit

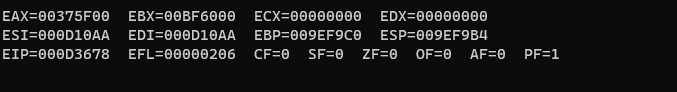
main ENDP

END main

Initially:



2nd last  




QUESTION 4:

INCLUDE Irvine32.inc

.data

N DWORD 5

fact DWORD 1

.code

main PROC

mov eax, fact ; eax with 1

mov ecx, N

L1:

mul ecx ; eax = eax \* ecx

loop L1

mov fact, eax

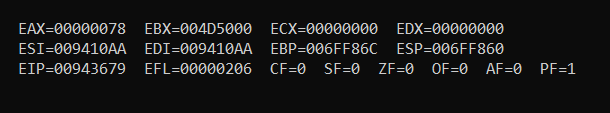
Done:

call DumpRegs

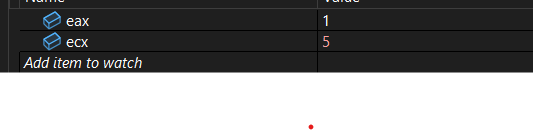
exit

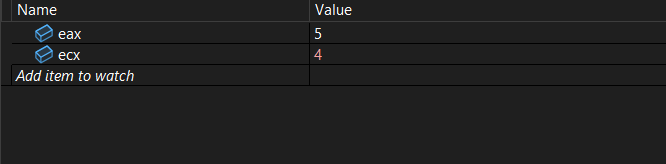
main ENDP

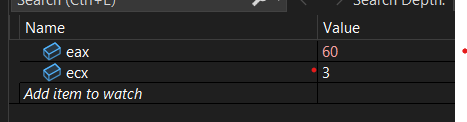
END main



Loop:





After 3 iteration  


QUESTION 5:

INCLUDE Irvine32.inc

.data

arr DWORD 24h, 93h, 22h, 54h, 10h, 0Fh

sum DWORD 0

.code

main PROC

mov esi, 0

mov eax, 0

mov ecx, LENGTHOF arr ; Loop no of iterations == number of elements of arr

L1:

mov ebx, DWORD PTR arr[esi\*4] ; Get the array element

add eax, ebx ; Add to sum

inc esi

loop L1

mov sum, eax

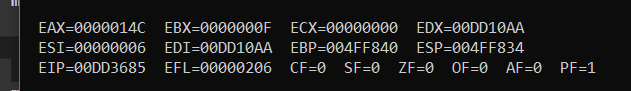
Done:

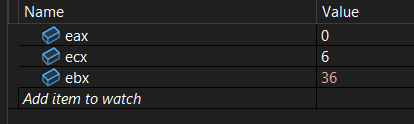
call DumpRegs

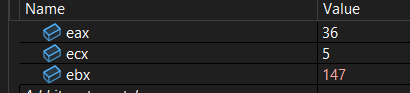
exit

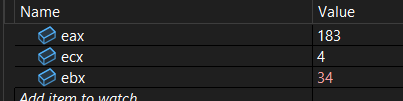
main ENDP

END main

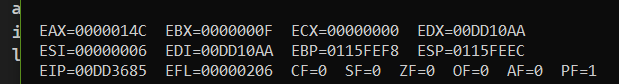






2nd loop  


After loop :



QUESTION 6:

INCLUDE Irvine32.inc

.data

arr BYTE 'A', 'B', 'C', 'D' ; Only one array

.code

main PROC

; Reverse the arra

mov esi, 0 ; Starting index

mov edi, 3 ; Ending index

mov ecx, 2 ; Only need to loop 2 times for swaping first half with second half of array

L1:

mov al, arr[esi] ; take value from start

mov bl, arr[edi] ; take value from end

mov arr[esi], bl ; Put end at start

mov arr[edi], al ; Put start at end

inc esi ; increment start

dec edi ; decrement end

loop L1 ; Repeat 2 times

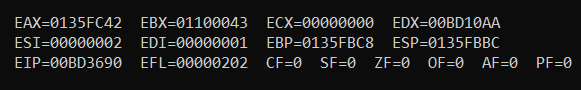
Done:

call DumpRegs ; Shows the reversed array in memory

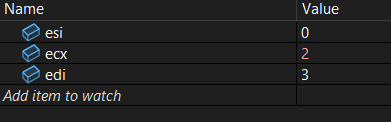
exit

main ENDP

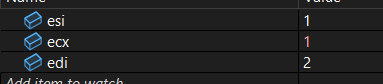
END main



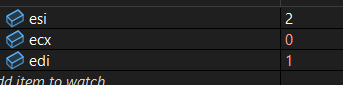
Initially



2nd loop



Outside loop



QUESTION 7:

INCLUDE Irvine32.inc

.data

arr BYTE 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19

even\_arr BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

odd\_arr BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

.code

main PROC

mov esi, 0

mov edi, 0

mov ebx, 0

mov ecx, 20

L1:

mov al, arr[esi]

mov dl, al

and dl, 1 ; Check last bit [ 1=odd , 0=even,]

cmp dl, 0

je EvenNumbers

mov odd\_arr[ebx], al

inc ebx

jmp Continued

EvenNumbers:

mov even\_arr[edi], al

inc edi

Continued:

inc esi

loop L1

Done:

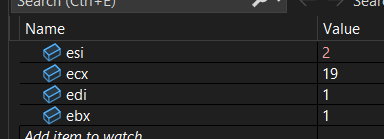
call DumpRegs

exit

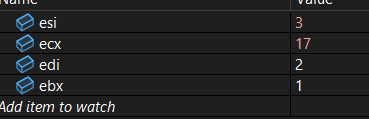
main ENDP

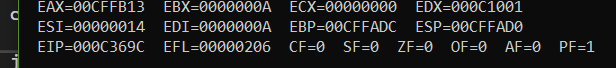
END main





2 values change this time





QUESTION 8:

INCLUDE Irvine32.inc

.data

num DWORD ? ; current number

.code

main PROC

mov eax, 0 ; accumulator

mov num, 1

mov ecx, 20 ; loop till 20 times

; nested loop

sumLoop:

mov edx, num

check\_if\_Even:

cmp edx, 2

jb checkremainder

sub edx, 2

jmp check\_if\_Even

checkremainder:

cmp edx, 0

jne skipAdd

add eax, num

skipAdd:

inc num

loop sumLoop

call DumpRegs

exit

main ENDP

END main

