# MUHAMMAD HASSAM KHAN

# SID: 11141

**LAB 8 COAL**

**Q1:**

**CODE:**

INCLUDE Irvine32.inc

.data

array2 byte 5,7,8,4,1,5,6,0 array3 byte ?

l1 dword ?

inseca byte ",",0 lm1 dword ?

msg1 byte ?

msg2 byte "before sorting: ",0 msg3 byte "after sorting: ",0

.code sortt proc mov eax,0 mov edx,0 mov esi,0

mov esi,OFFSET array2 mov l1,lengthof array2 mov ecx,0

mov ebx,l1 sub ebx,1

mov edx,offset msg2 call writestring call crlf

call crlf

.while ecx<l1 mov al,[esi+ecx] call writedec

mov edx,offset inseca abc:

cmp ebx,ecx je xyz

call writestring xyz:

inc ecx

.endw call crlf

call crlf call crlf mov ecx,0

.while ecx<l1 mov ebx,ecx add ebx,1

.while ebx<l1 mov al,[esi+ecx] mov ah,[esi+ebx] cmp al,ah

jae swap cmp al,ah jb dontswap swap:

mov msg1,al mov al,ah mov ah,msg1

mov [esi+ecx],al mov [esi+ebx],ah dontswap:

inc ebx

.endw inc ecx

.endw

mov edx,offset msg3 call writestring call crlf

call crlf mov eax,0 mov edx,0 mov esi,0

mov esi,OFFSET array2 mov l1,lengthof array2 mov ecx,0

mov ebx,l1 sub ebx,1

.while ecx<l1 mov al,[esi+ecx] call writedec

mov edx,offset inseca abc1:

cmp ebx,ecx je xyz1

call writestring xyz1:

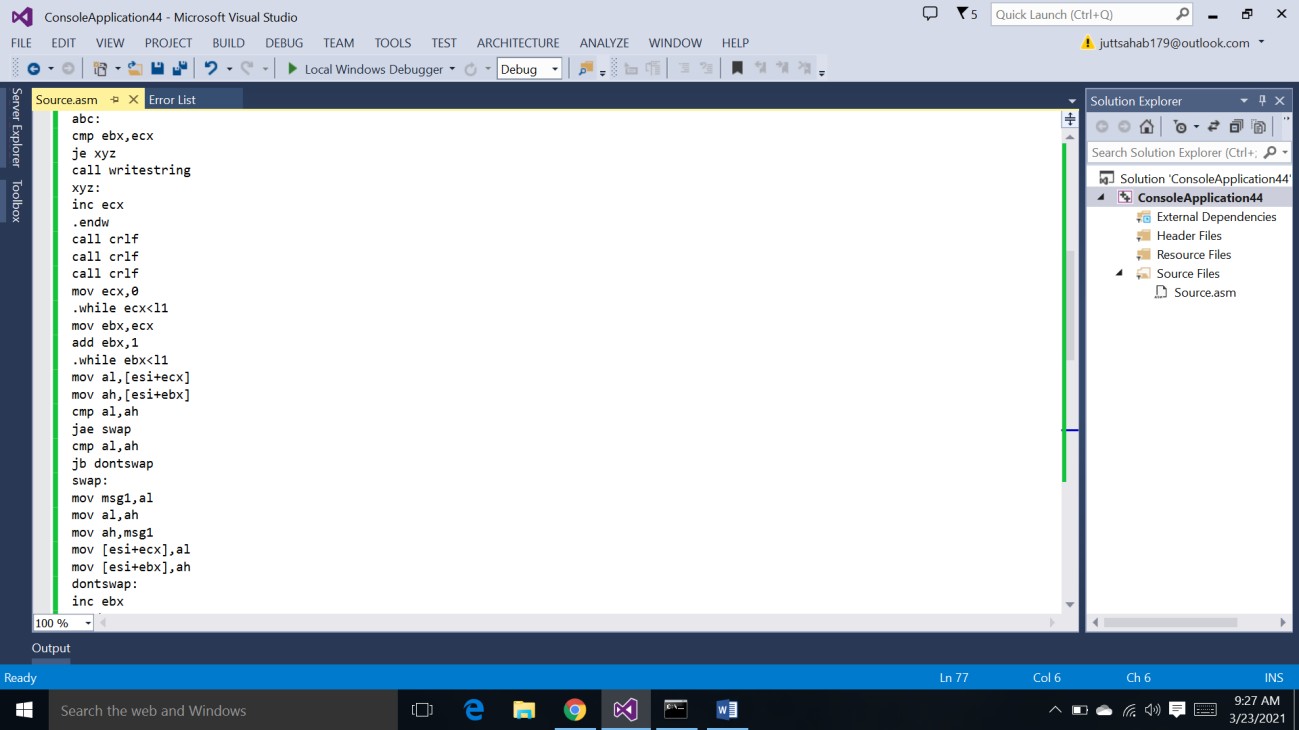
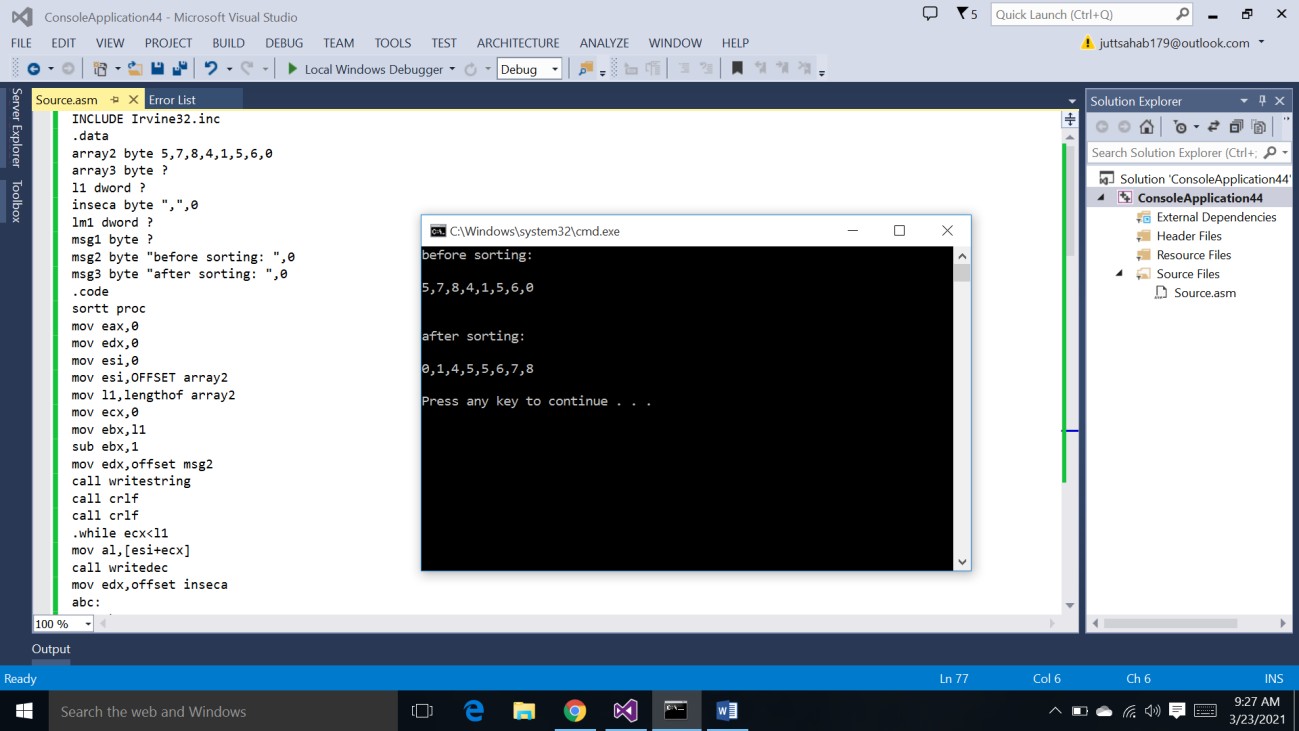
inc ecx

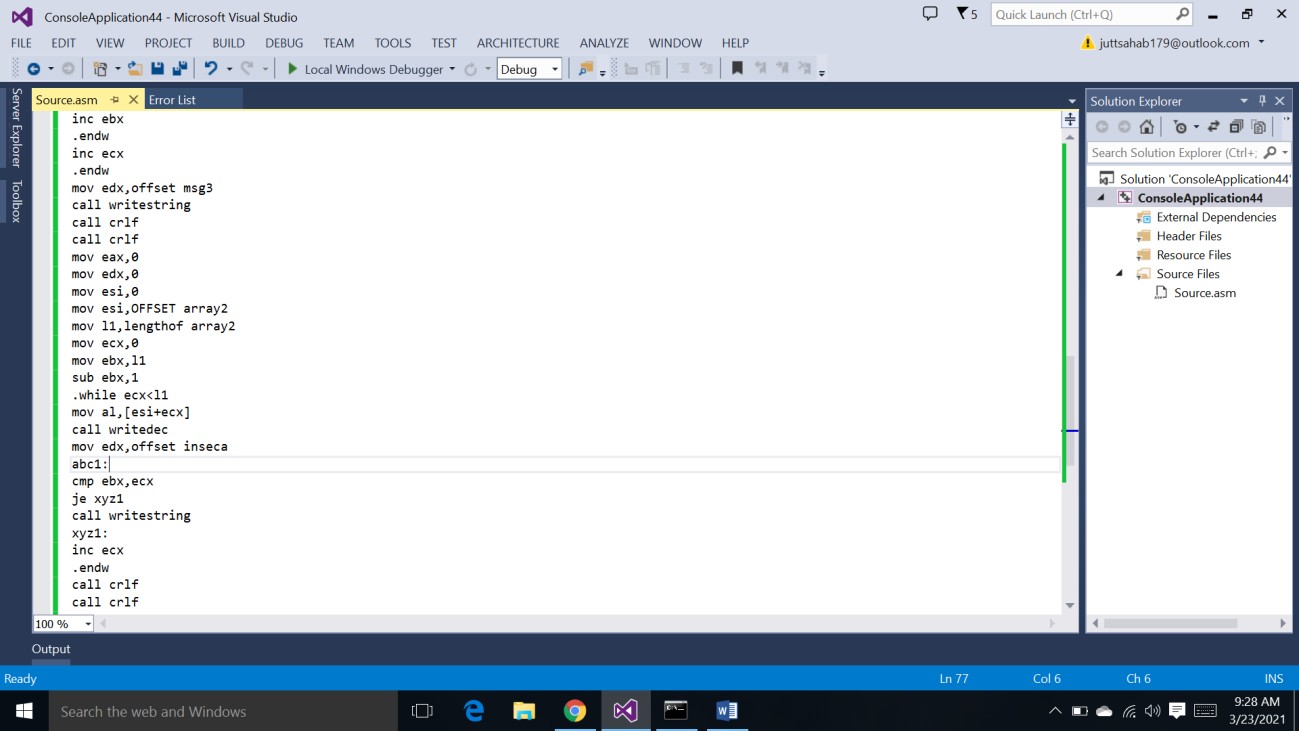
.endw call crlf call crlf ret

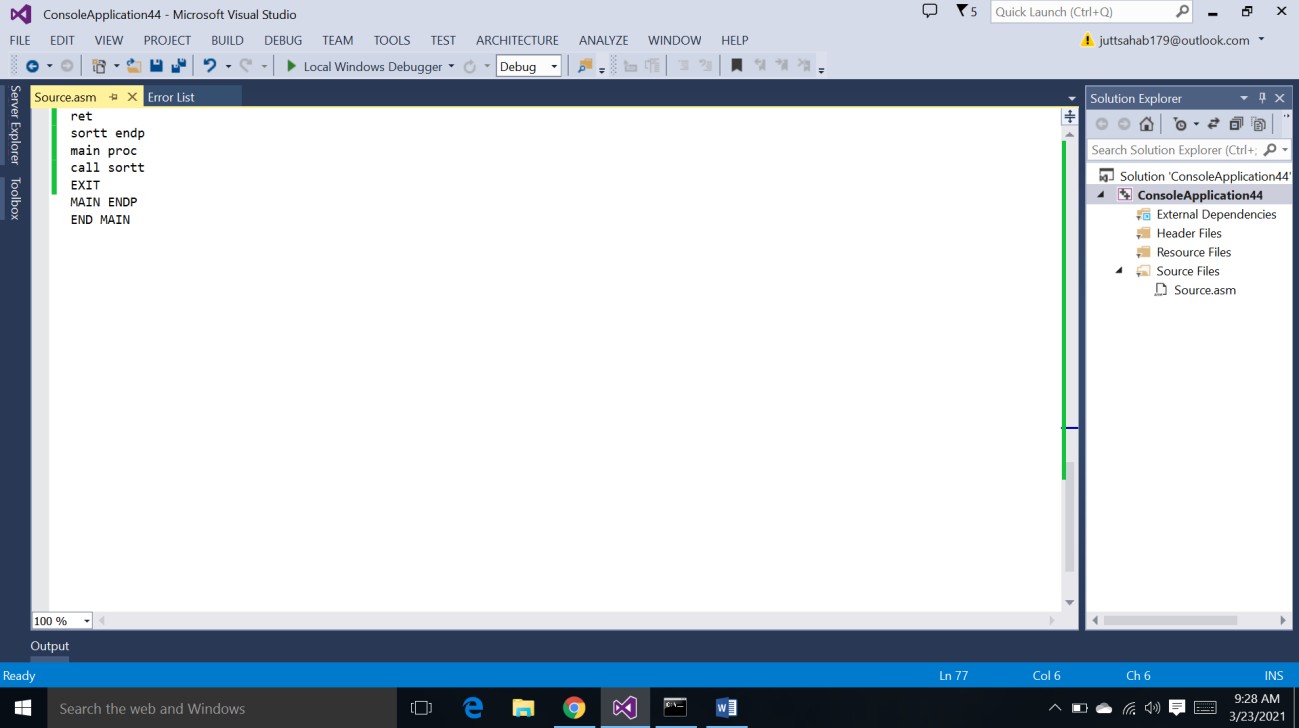
sortt endp main proc call sortt EXIT

MAIN ENDP END MAIN

# SCREENSHOT:







**Q3:**

**CODE:**

INCLUDE Irvine32.inc

.data n=1000

arr dword n Dup(?) evearr dword n Dup(?) oddarr dword n Dup(?) eve dword 4

odl dword 4 temp dword ?

msg1 byte "value to enter in array : ",0 msg2 byte "even numbers",0

msg3 byte "odd number",0 value byte "Value: ",0 len dword ?

d dword 2

multiplier dword 4

check dword 0 limit dword ?

.code main proc

mov edx,offset msg1 call writestring call readdec

mov limit,eax mul multiplier mov len,eax mov ecx,4

mov eax,0

.while ecx<=len

mov edx,offset value call writestring call readdec

mov [arr+ecx],eax add ecx,4

.endw call crlf mov ecx,4

.while ecx<=len mov eax,[arr+ecx] mov ebx,eax

mov edx,0 div d

mov temp,ecx cmp edx,check je ee

jne oo B:

mov ecx,temp add ecx,4

.endw

mov edx,offset msg2 call writestring call crlf

mov ecx,4

mov eax,0

.while ecx<eve

mov eax,[evearr+ecx] call writedec

call crlf add ecx,4

.endw

mov edx,offset msg3 call writestring call crlf

mov ecx,4 mov eax,0

.while ecx<odl

mov eax,[oddarr+ecx] call writedec

call crlf add ecx,4

.endw exit ee:

mov ecx,eve

mov [evearr+ecx],ebx add eve,4

jmp B oo:

mov ecx,odl

mov [oddarr+ecx],ebx add odl,4

jmp B

MAIN ENDP END MAIN

# SCREENSHOT:

