# QUESTION 1:

# TASK 2:

org 0x0100

section.data

grades dw 69,87,96,45,75

size db 5

section.text

mov ax,0

outer\_loop:

mov dx, ax

mov al,0

mov cx,ax

inner\_loop:

inc cx

cmp cx, size

je inner\_done

mov bx,cx

mov bl,[grades+bx]

mov bh,al

cmp bl,bh

jge continue\_loop

mov al,bl

mov dx,cx

continue\_loop:

jmp inner\_loop

inner\_done:

mov si,ax

mov di,dx

call swap\_elements

inc ax

cmp ax,size

jne outer\_loop

mov ah, 0x4c

int 0x21

swap\_elements:

mov bx,ax

mov si,[grades+bx]

mov bx,dx

mov di,[grades+bx]

mov bx,ax

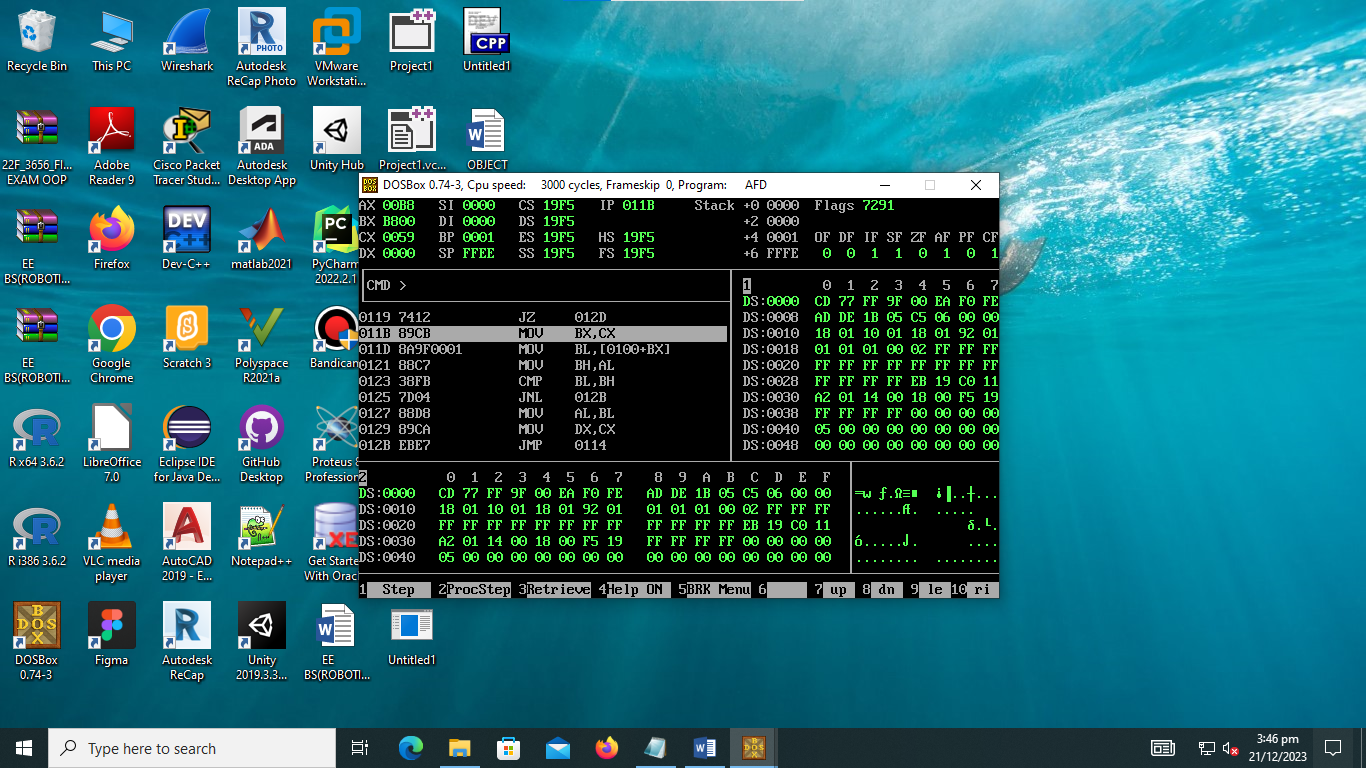
mov [grades+bx],di

mov bx,dx

mov [grades+bx],si

ret

# OUTPUT:



# Question 2:

# TASK 2:

org 0x0100

jmp start

clrscr:

push es

push ax

push cx

push di

mov ax, 0xb800

mov es,ax

mov di,0

mov ax,0x0720

mov cx,2000

cld

rep stosw

pop di

pop cx

pop ax

pop es

nextchar:

add di,2

cmp di,4000

jne nextchar

ret

drawTriangle:

push bp

mov bp,sp

mov cx, [bp+4]

mov bx, [bp+6]

l1:

push cx

inc bx

mov cx,bx

l2:

mov ax,cx

imul ax,50

add ax,bx

imul ax,3

mov ah,2

mov dl,'\*'

int 21h

loop l2

mov ah,9

mov dx, data

int 21h

pop cx

loop l1

pop bp

ret

start:

mov cx,8

mov bx,0

mov sp, 0x0100

push bx

push cx

call clrscr

call drawTriangle

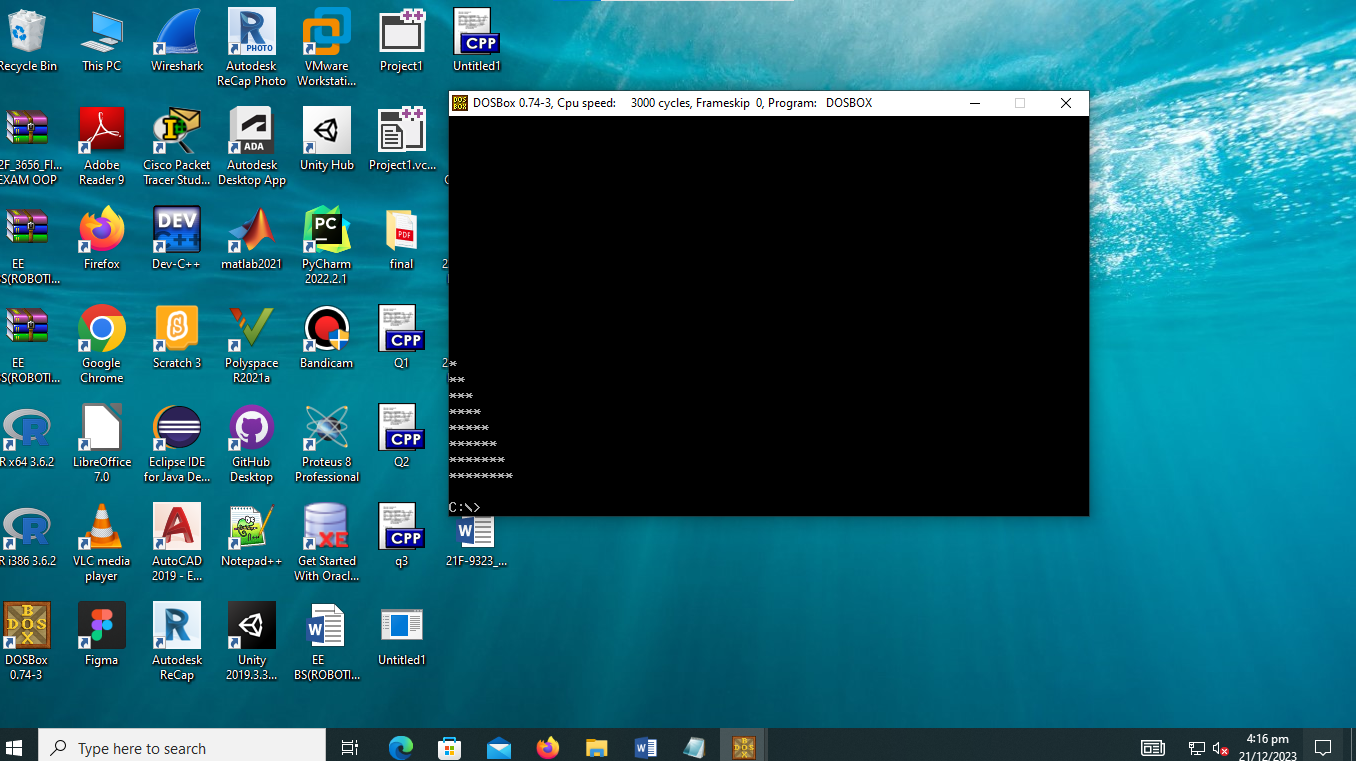
add sp,4

mov ah,4Ch

int 21h

section.data

data db 10,'$'

OUTPUT:  


QUESTION 3:

org 0x0100

jmp start

kbisr:

push ax

push es

mov ax, 0xb800

mov es,ax

in al,0x060

cmp al,0x0f

jne nomatch

cmp al,0x039

jne nomatch

nomatch:

mov al,0x20

out 0x20,al

pop es

pop ax

iret

start:

xor ax,ax

mov es,ax

cli

mov word [es:0\*4],kbisr

mov [es:0\*4+2],cs

sti

l1

jmp l1

# OUTPUT: