

Arab American University

Faculty of Engineering and Information Technology

Assembly lab

Prepared by:Suha Waleed Alardah

ID:201811033

To:Dr.Tareq Zanoon

To:Mrs.NOOR

***Introduction:*Addressing modes are an aspect of the instruction set architecture in most central processing unit (CPU) designs. The various addressing modes that are defined in a given instruction set architecture define how the machine language instructions in that architecture identify the operand(s) of each instruction. However, in 8086 microprocessor family there exits seven addressing mode.In this experiment we will use different addressing modes in the same program.**

*Objectives:*

* Using different addressing modes in the same assembly code.
* Introduce the seven addressing modes and the different.

## Type of addressing mode :

Based register : one operand is a register while the other one is one of based registers (BX,BP) that  contains the offset of the data.

Index register : one operand is a register while the other one is one of indexes registers (SI,DI) contains the offset of the data.

 Register operand: both operands are registers.

Immediate: one operand is a register while the other one is immediate data.

Direct  : one operand is a register while the other one is the offset address of the data.

Register indirect: one operand is a register while the other one is other register contains the offset of the data.

Index-Based register : one operand is a register while the other one is the summation of an index register with base register that indicates  the offset of the data.

**Task1**: write an assemply language program that allows the user to enter a group of characters .The entry process stops when the user presses enter.

dosseg

.model small

.data

.code

main:

rep :

mov ah, 1

int 21h

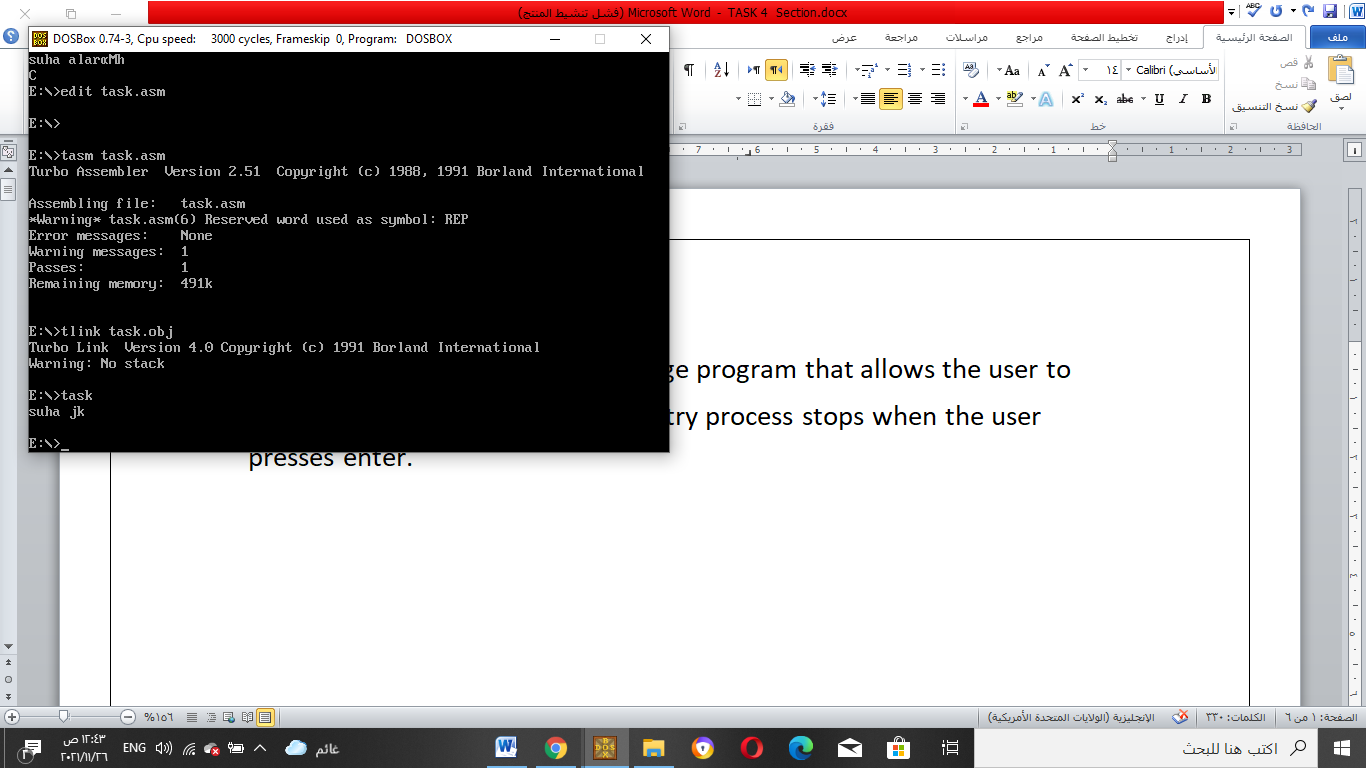
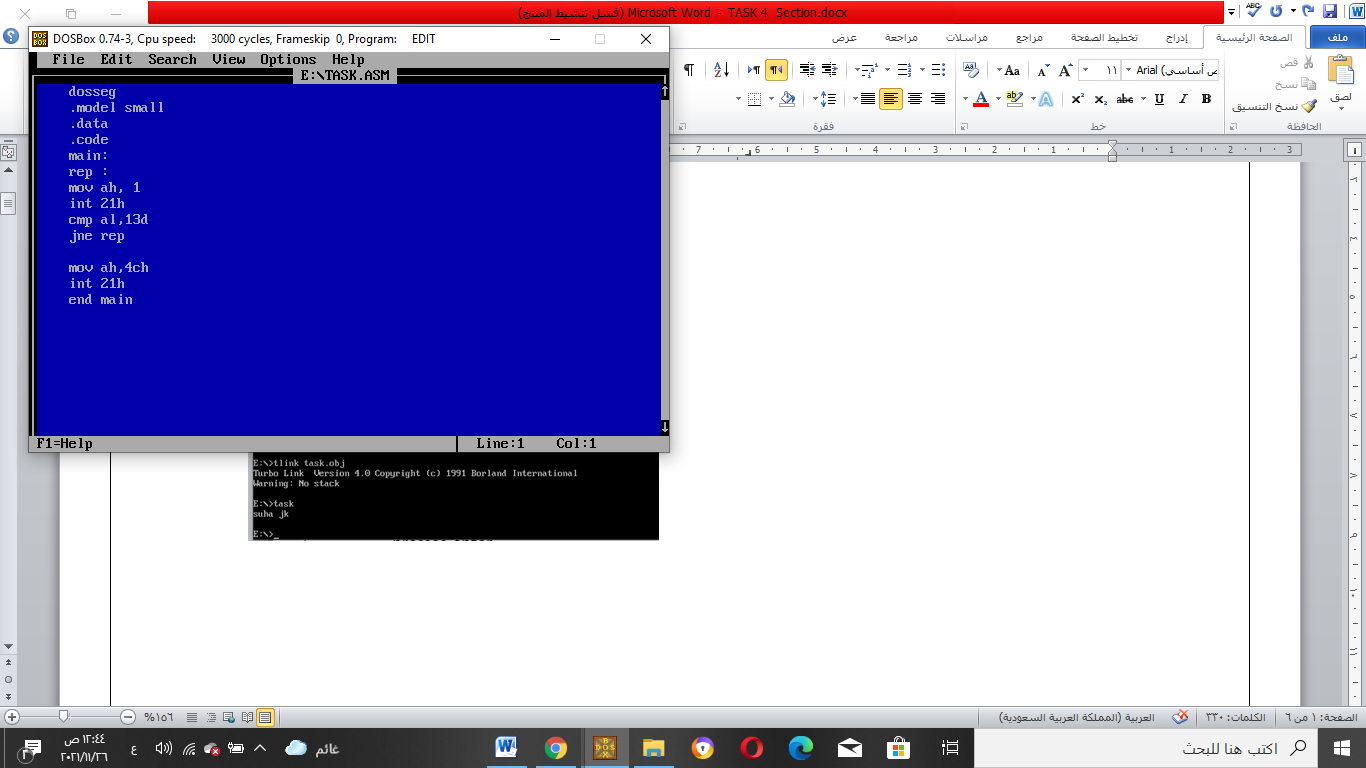
cmp al,13d

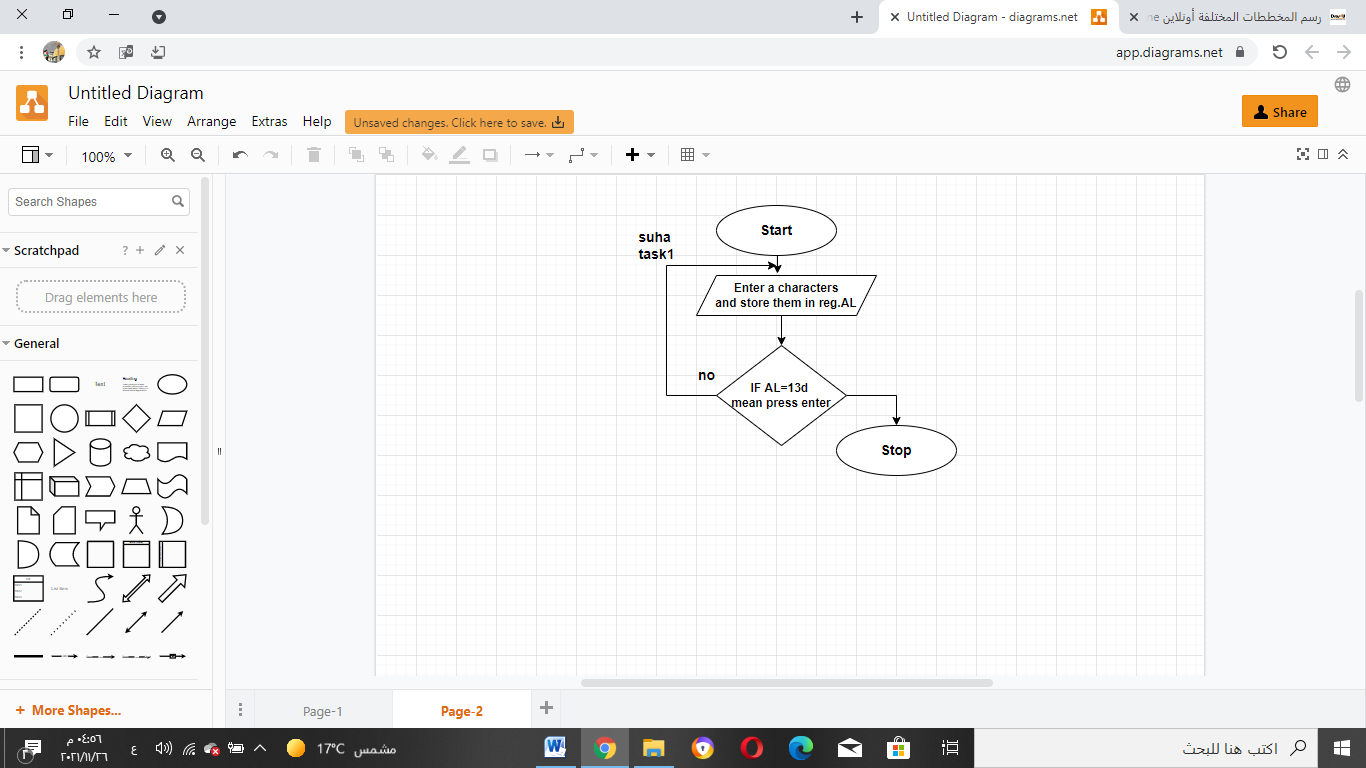
jne rep

mov ah,4ch

int 21h

end main





**Task2:** print the nimber of characters enter by the user .assume to be less than 10

dosseg

.model small

.data

.code

main:

mov ax,@data

mov ds ,ax

mov si,400

mov bl,0

rep:

mov ah,1

int 21h

inc bl

cmp al,13d

jne rep

dec bl

mov dl,bl

add dl,30h

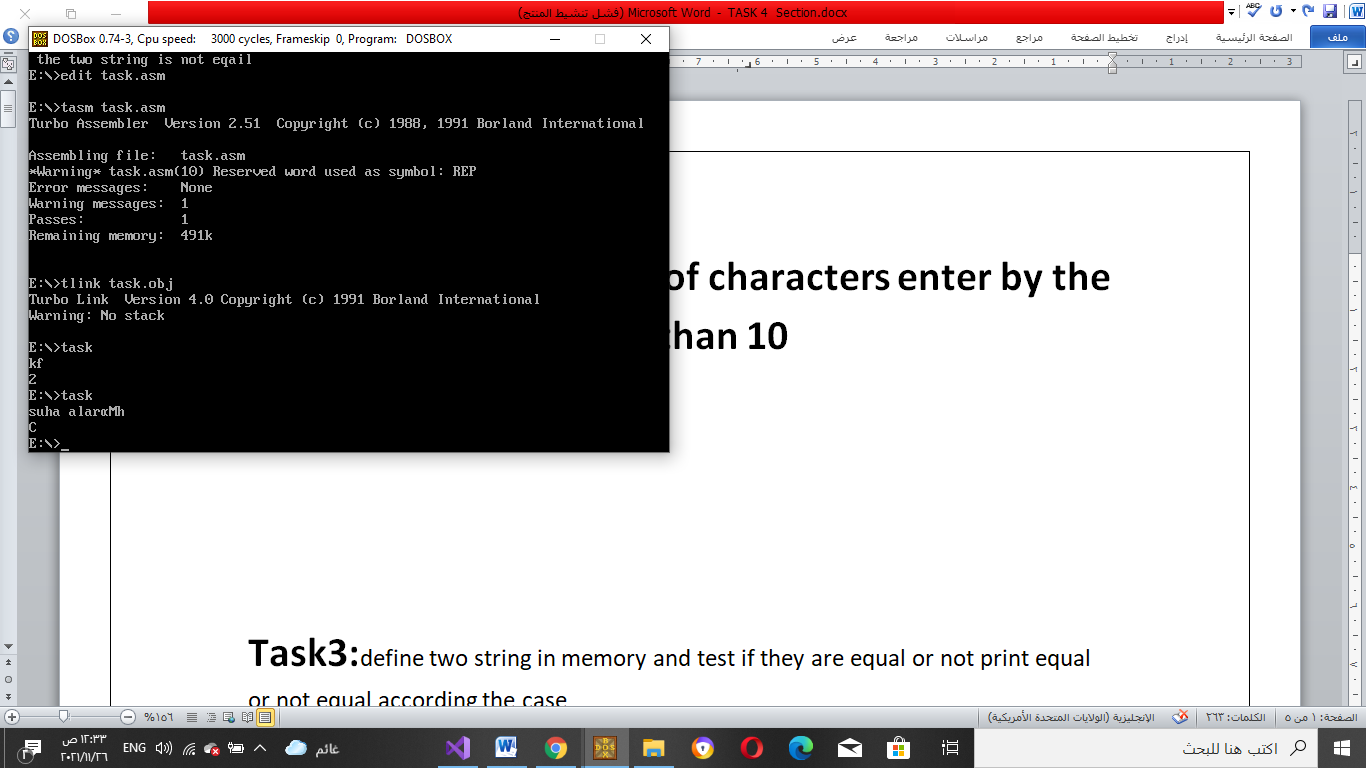
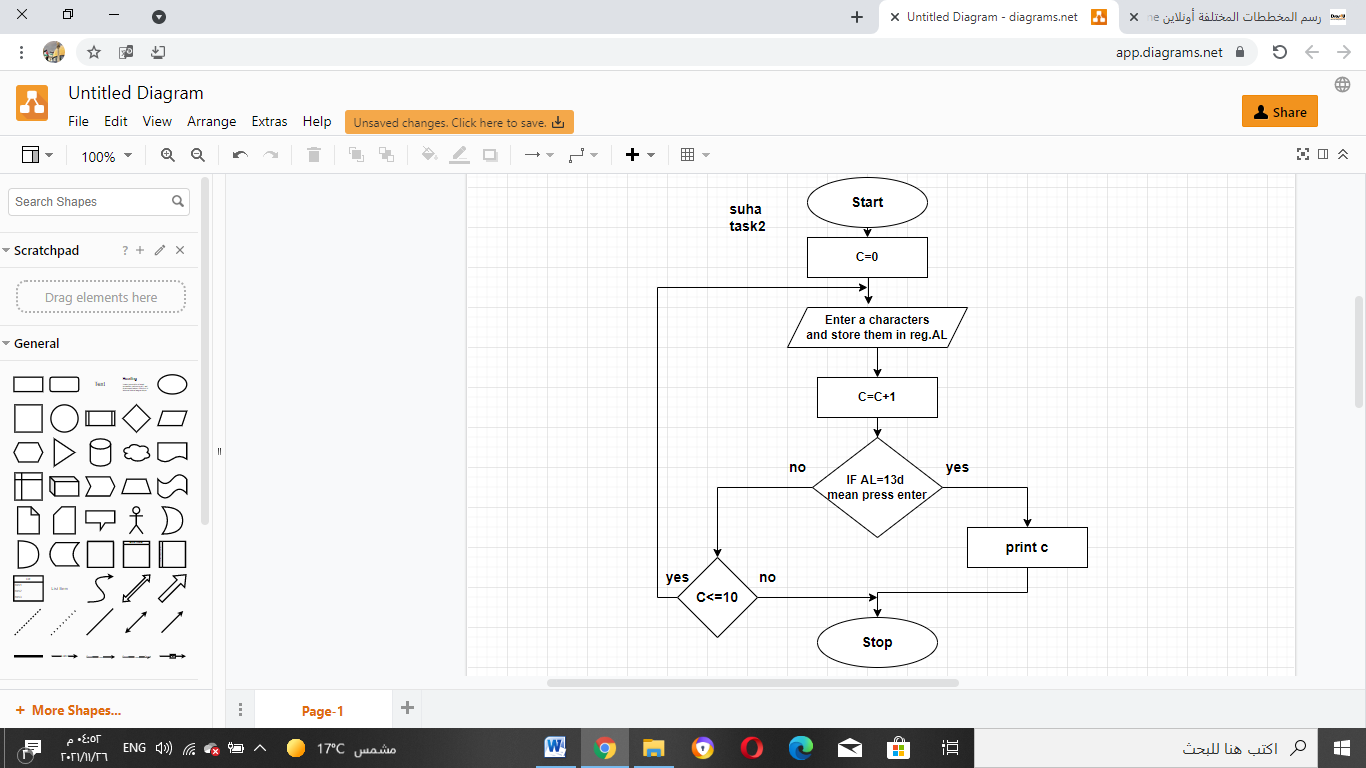
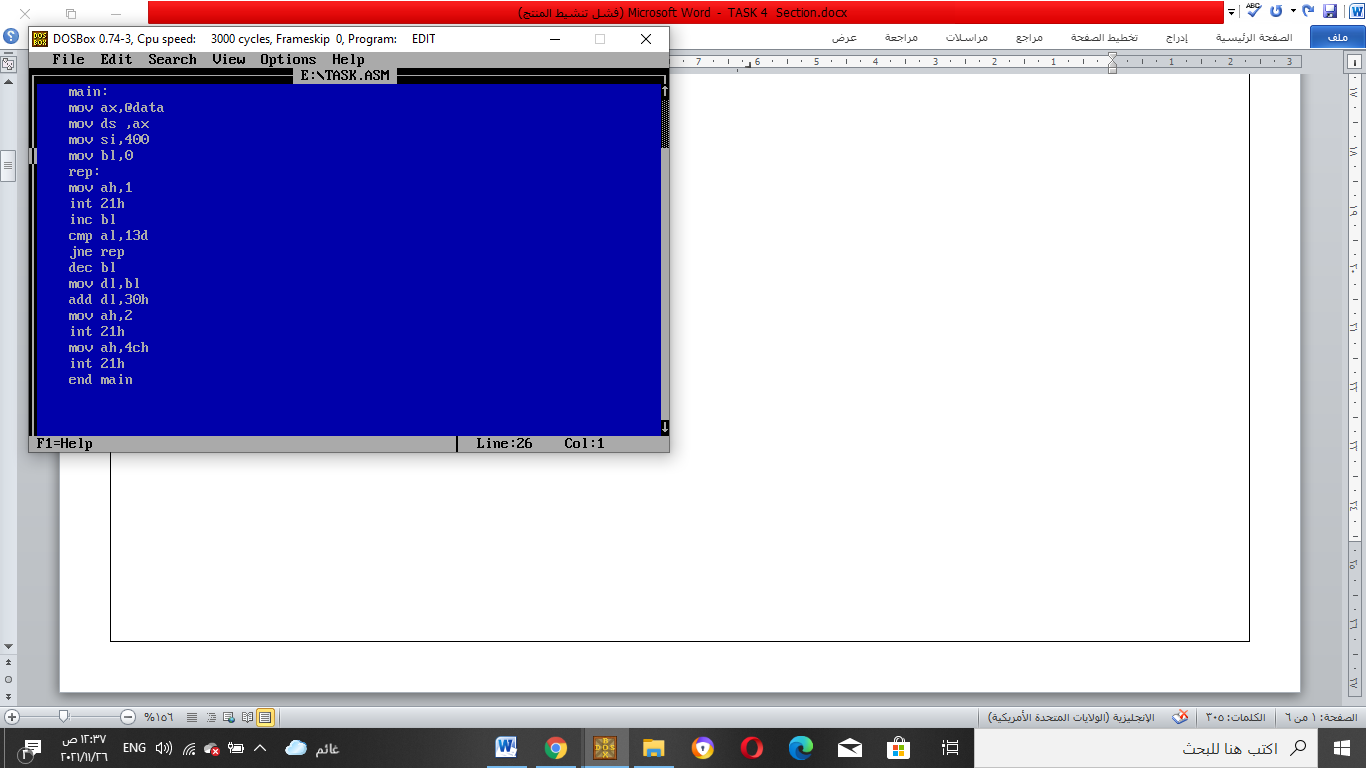
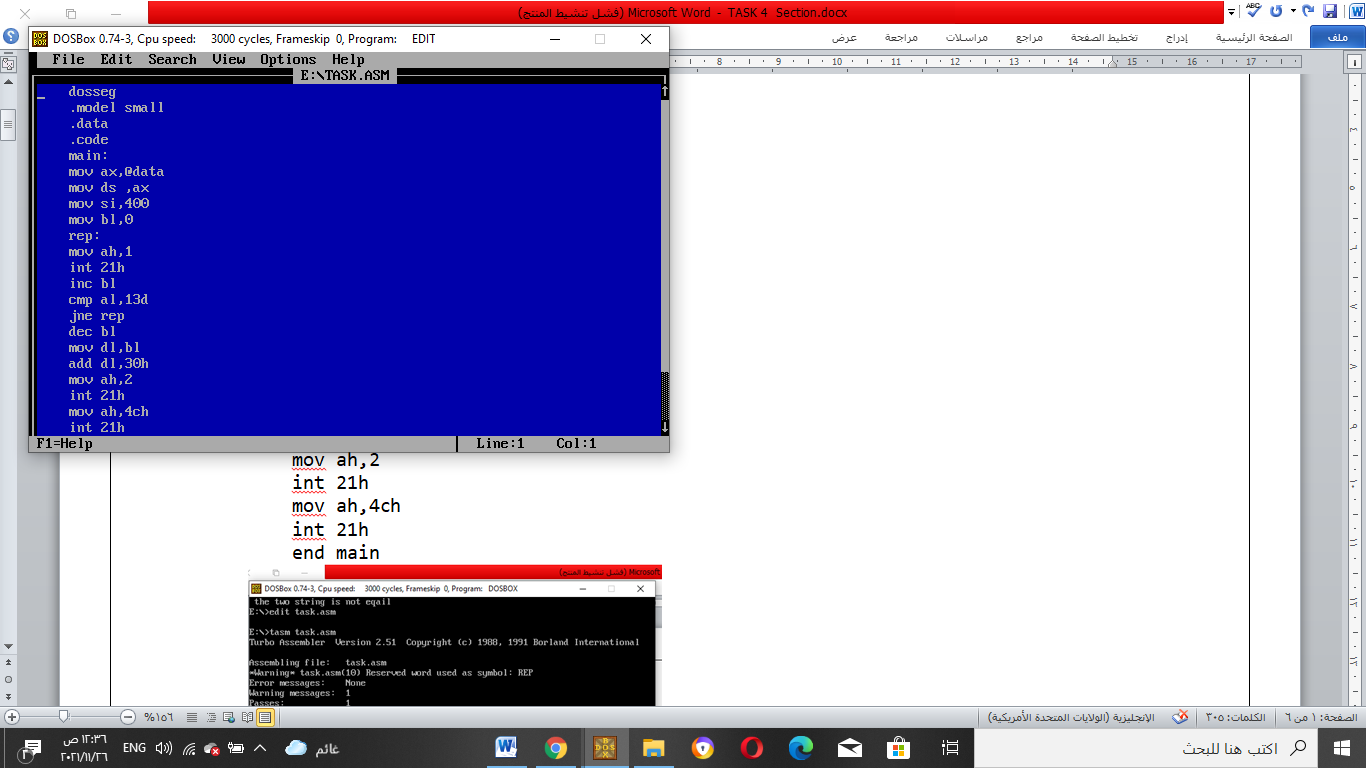
mov ah,2

int 21h

mov ah,4ch

int 21h

end main



**Task3:**define two string in memory and test if they are equal or not print equal or not equal according the case

dosseg

.model small

.data

msg1 db "suha ",10,13,"$"

msg2 db "suha2 ",10,13,"$"

first db " the two string is not eqail$"

second db "the two string is eqail$"

.code

main :

mov ax,@data

mov ds,ax

mov bx,0

mov cx,0

mov di,offset msg1

mov si,offset msg2

loop1:

mov bl,[di]

mov cl,[si]

cmp cl,bl

jnz loop2

cmp bl,"$"

jz loop3

inc si

inc di

jmp loop1

loop2:

mov ah,9

mov dx,offset first

int 21h

jmp loop4

loop3:

mov ah,9

mov dx, offset second

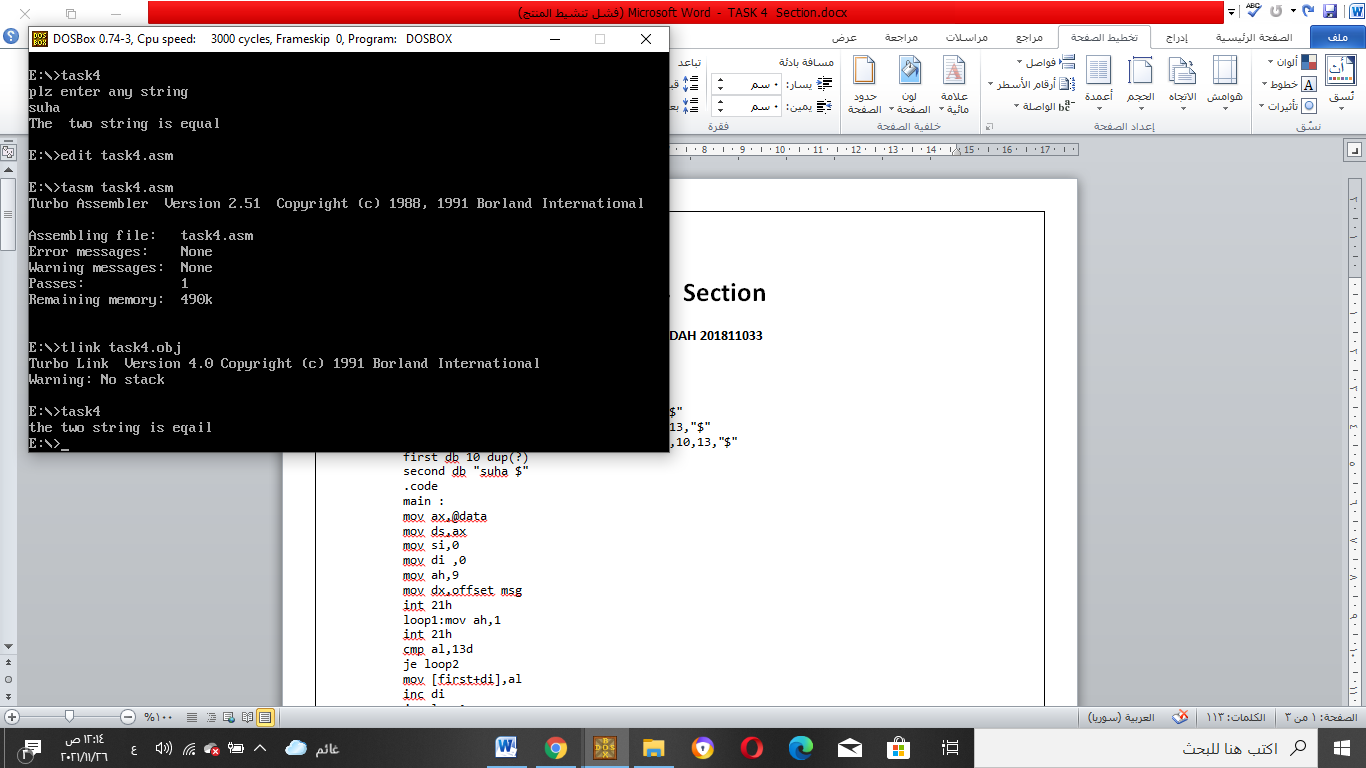
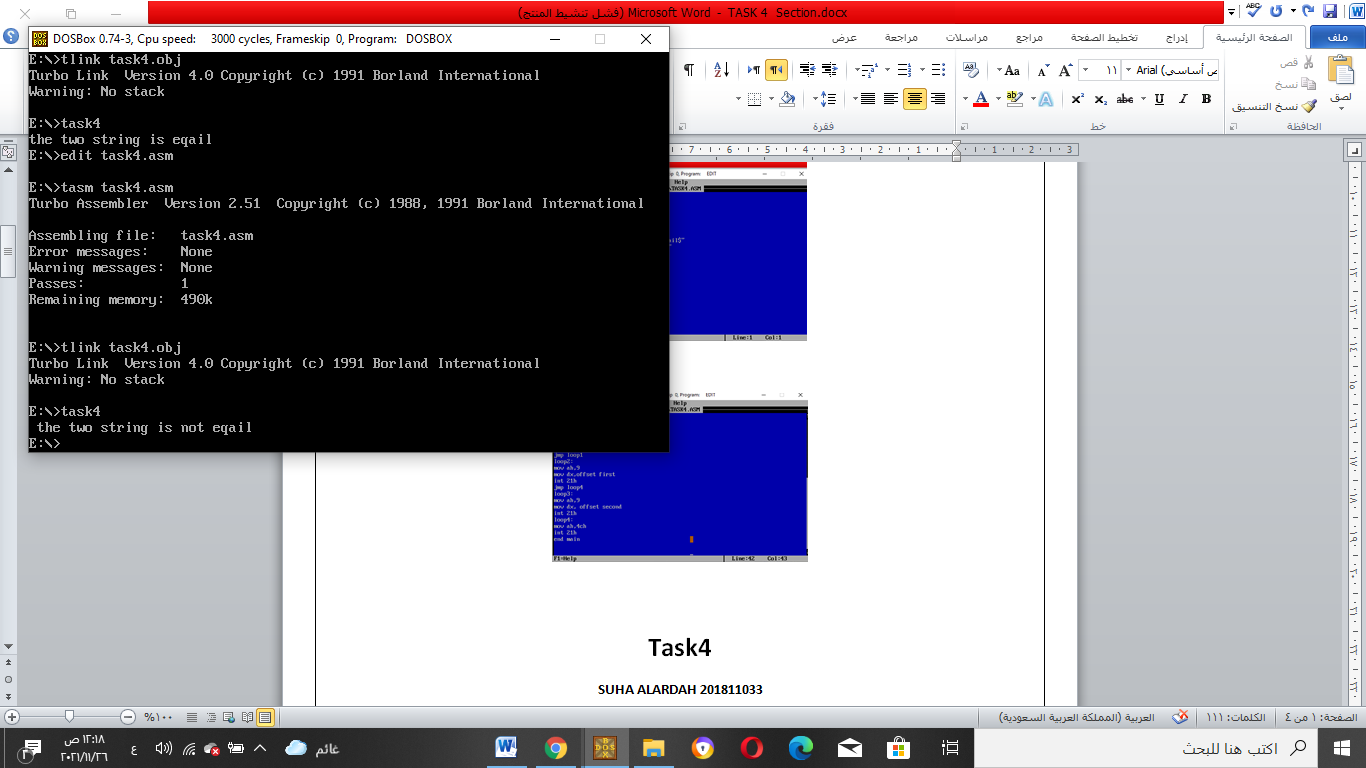
int 21h

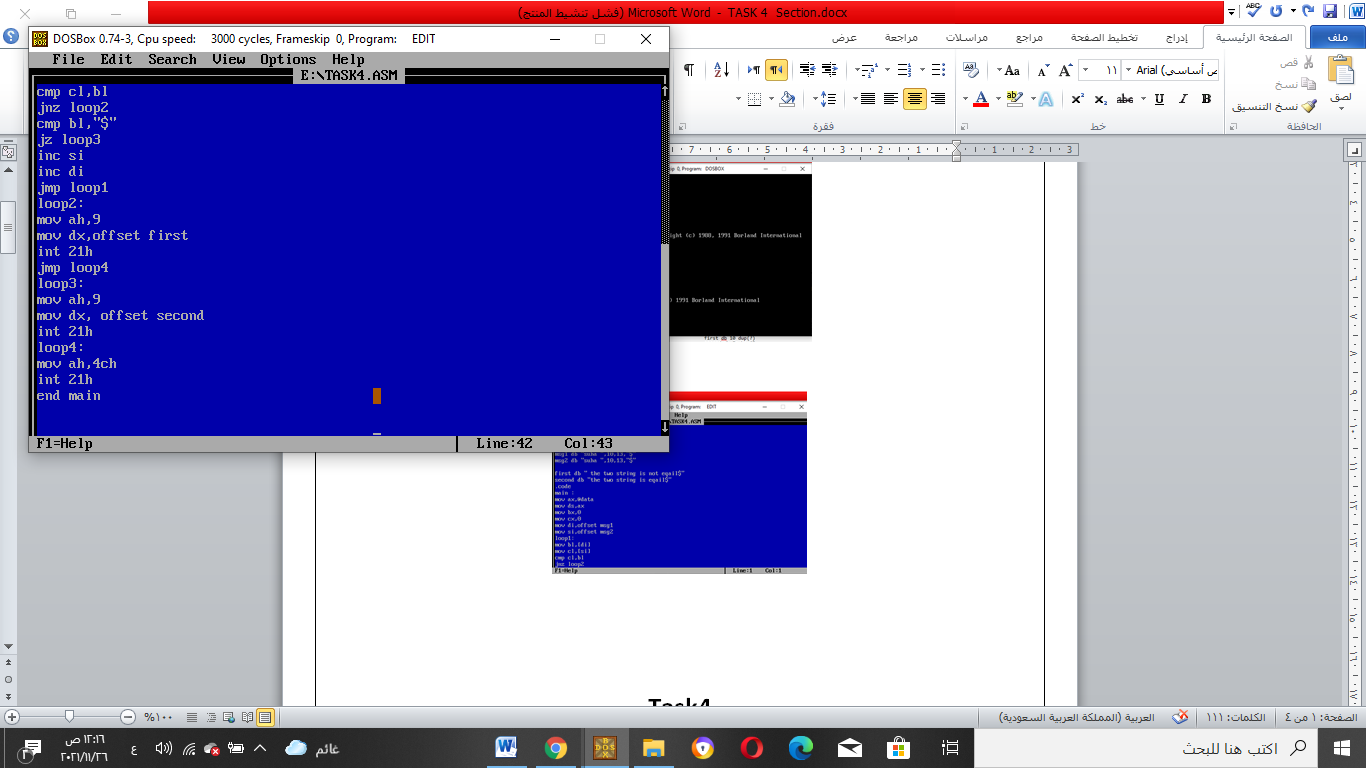
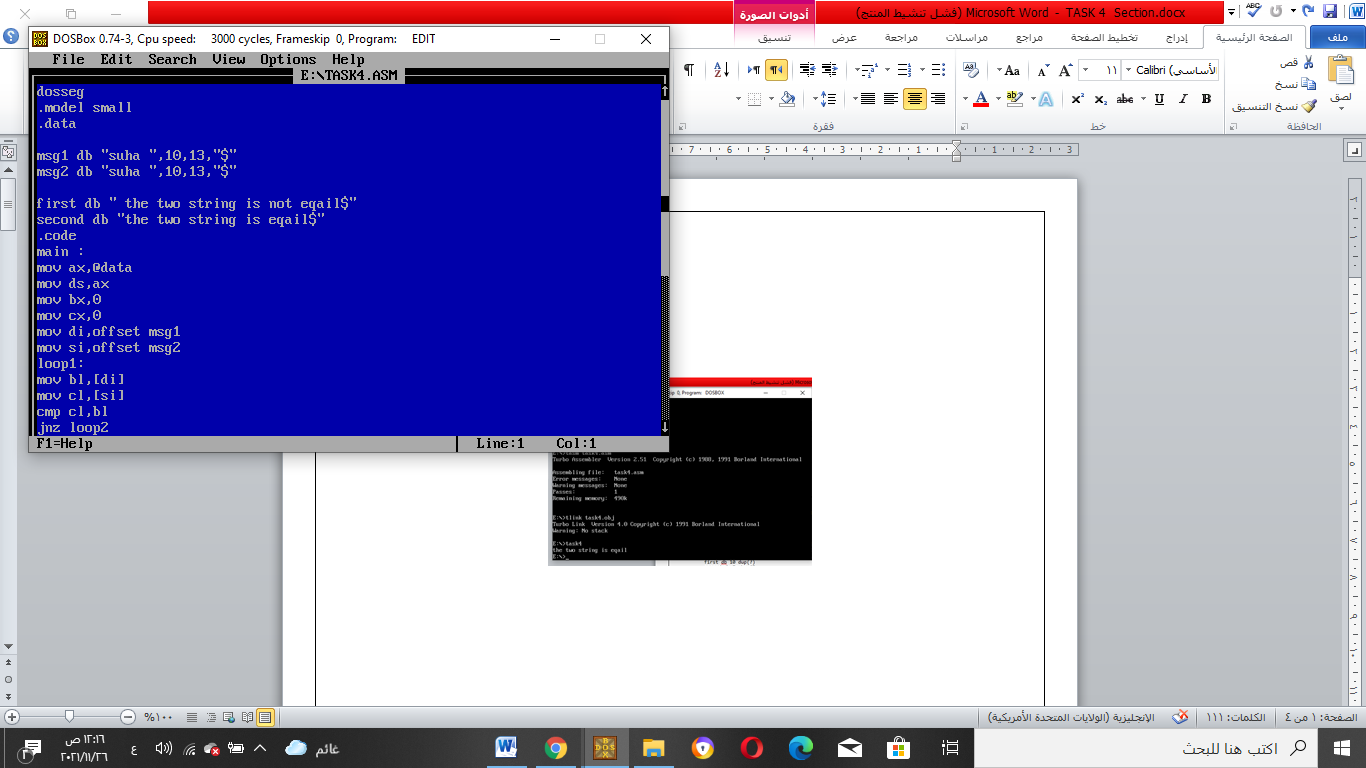
loop4:

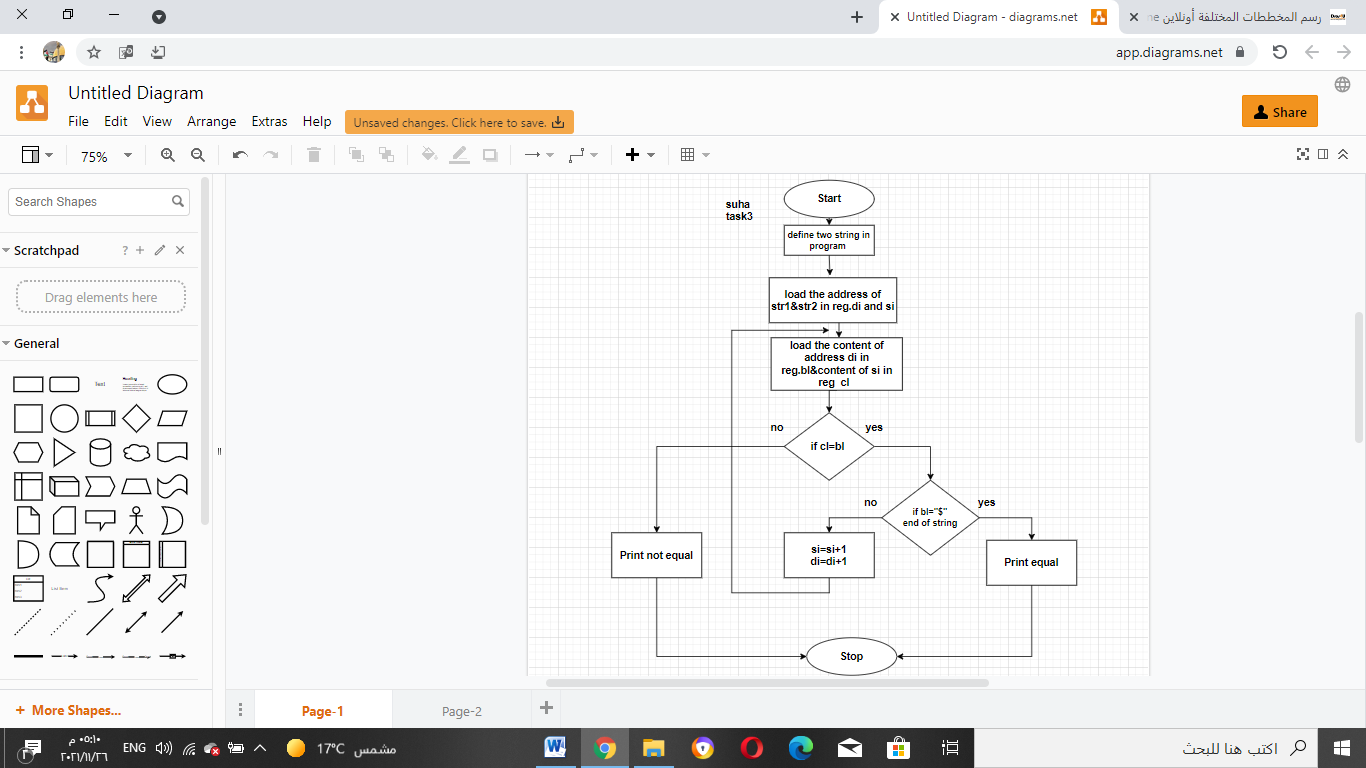
mov ah,4ch

int 21h

end main







**Task4:**Modify program 3 such that one string is defined in memory while the other is enterd after the keyboard print equal or not equal depanding on the case .

dosseg

.model small

.data

msg db "plz enter any string ",10,13,"$"

msg1 db "The two string is equal",10,13,"$"

msg2 db "The two string is not eqaul",10,13,"$"

first db 10 dup(?)

second db "suha $"

.code

main :

mov ax,@data

mov ds,ax

mov si,0

mov di ,0

mov ah,9

mov dx,offset msg

int 21h

loop1:mov ah,1

int 21h

cmp al,13d

je loop2

mov [first+di],al

inc di

jmp loop1

loop2:

mov cl,[second+si]

cmp [first+si],cl

jne loop4

inc si

cmp si,di

jne loop2

loop3:

mov ah,9

mov dx,offset msg1

int 21h

jmp finel

loop4:

mov ah,9

mov dx,offset msg2

int 21h

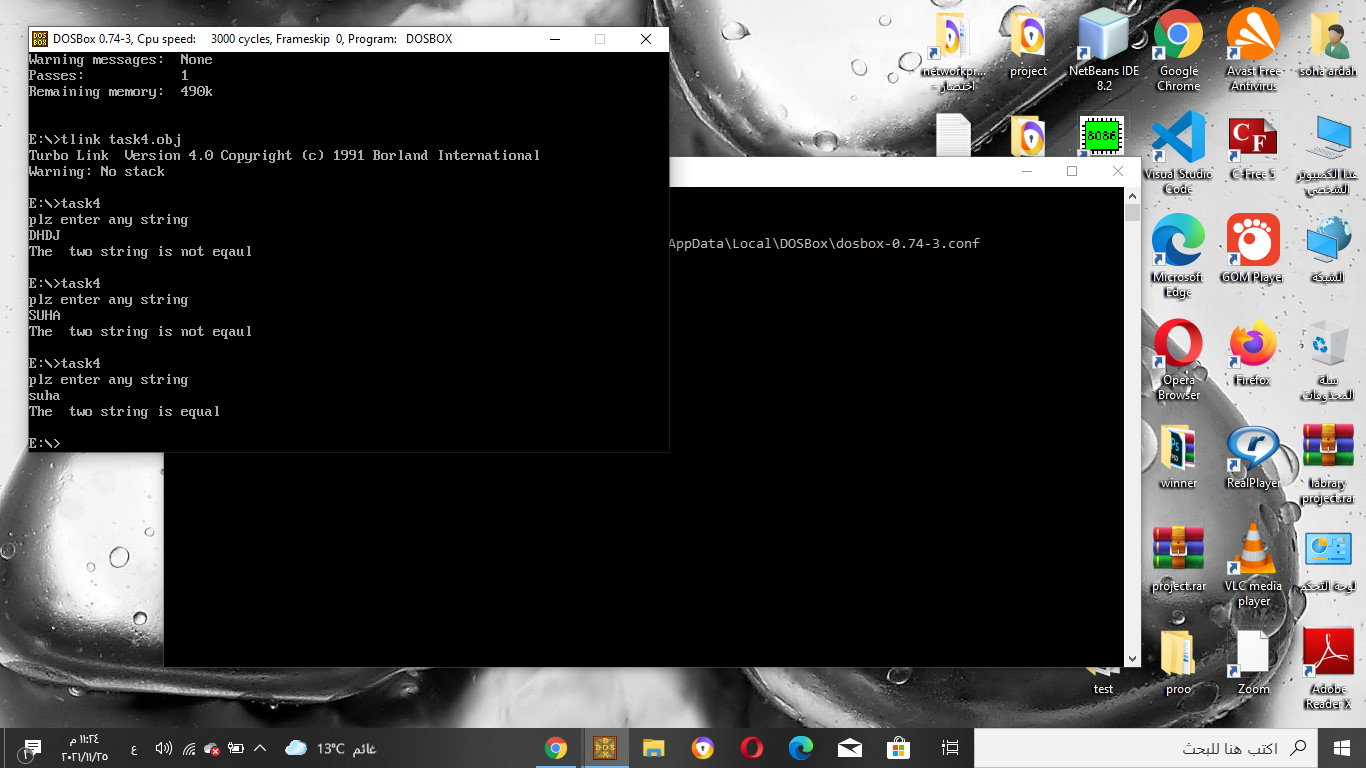
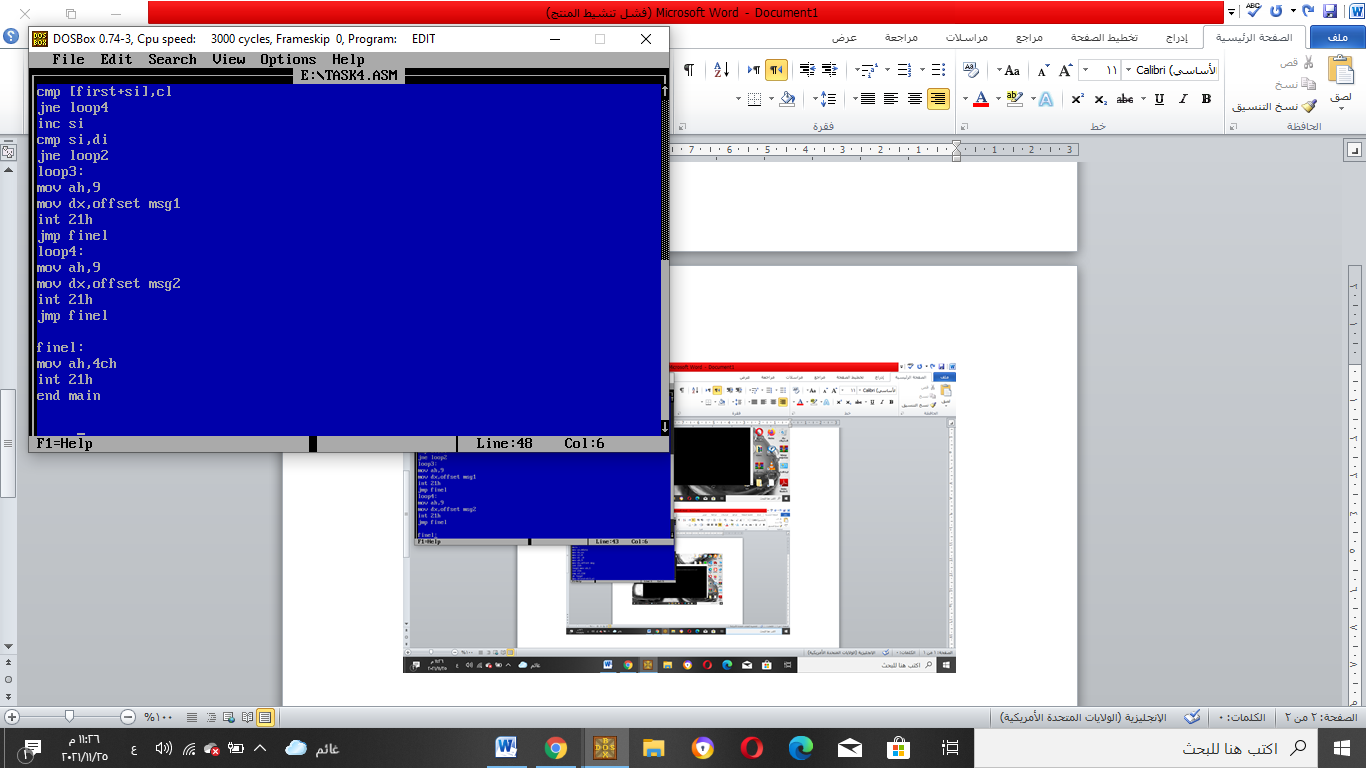
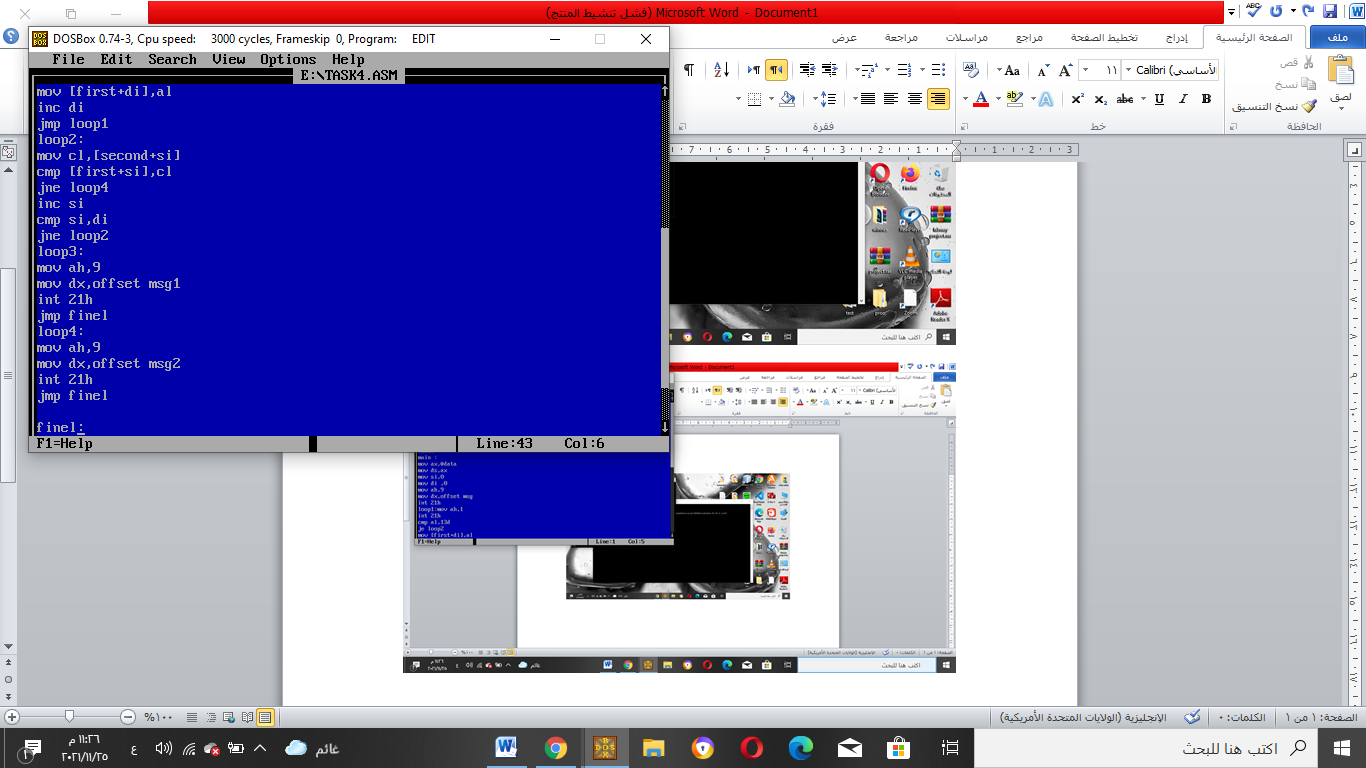
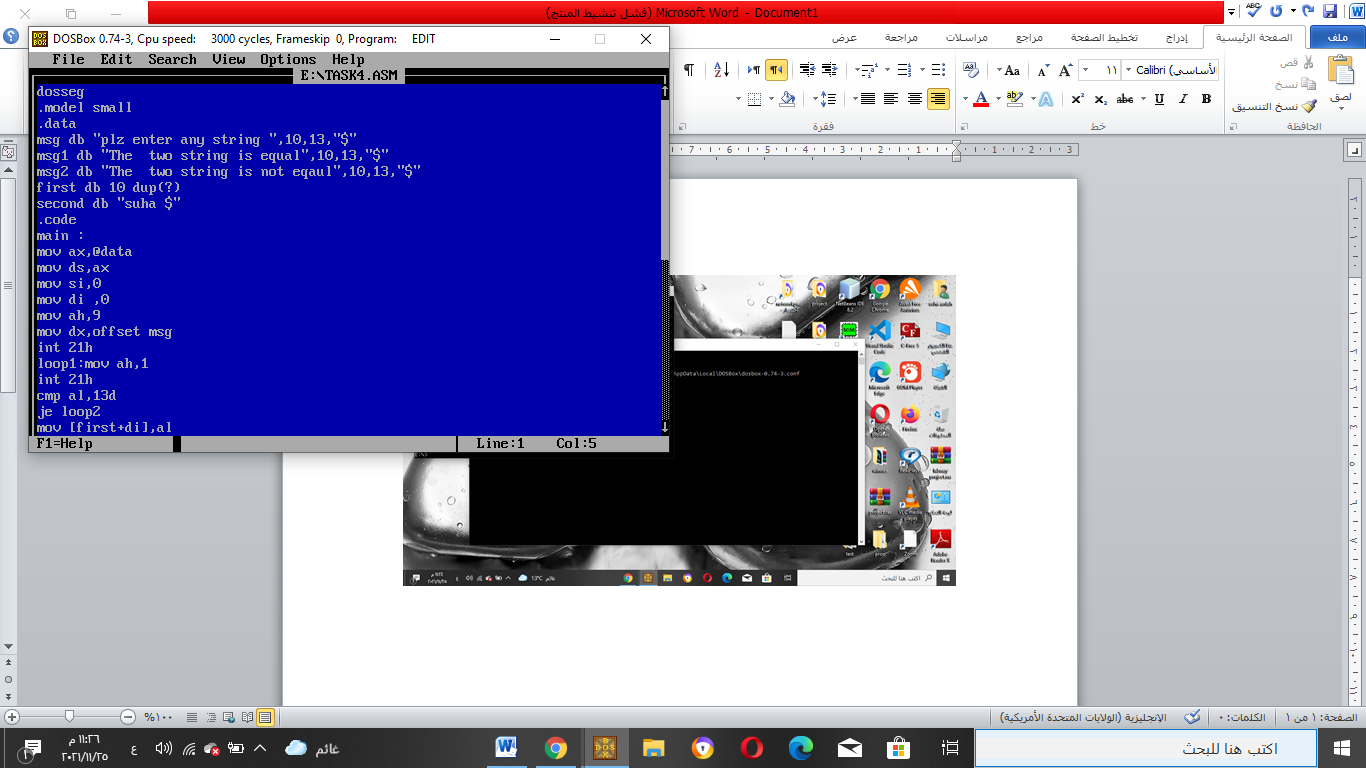
jmp finel

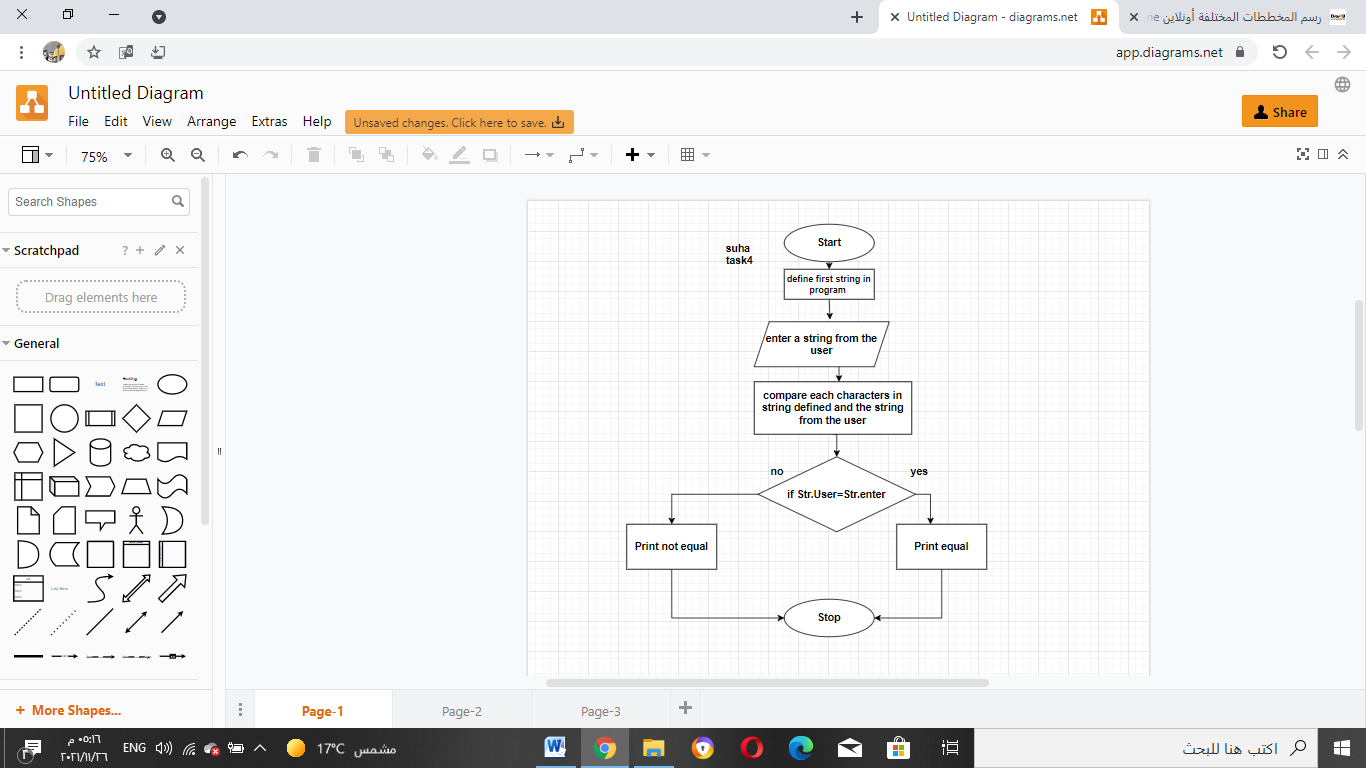
finel:

mov ah,4ch

int 21h

end main





*Conclusion:*

In this lab we use different address mode

 Immediate (MOV AH,1),  register (MOV DX,SI), index indirect (MOV AL,[SI]) .

Finally Addressing modes are the way CPU access the main memory. 8086 family enable seven different addressing modes which makes assembly programming easily and more smoothly.