

**Arab American University**

**Faculty of Engineering and Information Technology**

**Assembly lab**

**Lab7:Stacks and Subroutines**

**Prepared by:Suha Waleed Alardah**

**ID:201811033**

**To:Dr.Tareq Zanoon**

**To:Mrs.NOOR**

**Introduction :**

A subroutine is a block of code that is called from different places from within a main program or other subroutines. l Saves code space in that the subroutine code does not have to be repeated in the program areas that need it; l Only the code for the subroutine call is repeated. l A subroutine can have l parameters that control its operation l local variables for computation. l A subroutine may pass a return value back to the caller. l Space in data memory must be reserved for parameters, local variables, and the return value.

And the stack is part of memory used for temporary storage of addresses

and data. The SP register points to the top of the stack at any given time

The PUSH instruction is used to place values on the stack and the POP

instruction is used to remove values from the stack.

The SS and SP registers point to the top of the stack like this: SS:[SP].

So the SS register is the segment and the SP register contains the offset.

When you push a word (2 bytes) for example, SP will be decreased two

then the word will be stored at SS:[SP].

Task1: Write assembly to set all flag bit (td)

dosseg

.model small

.data

.code

main :

mov ax,@data

mov ds,ax

mov ax,0ffffh

push ax

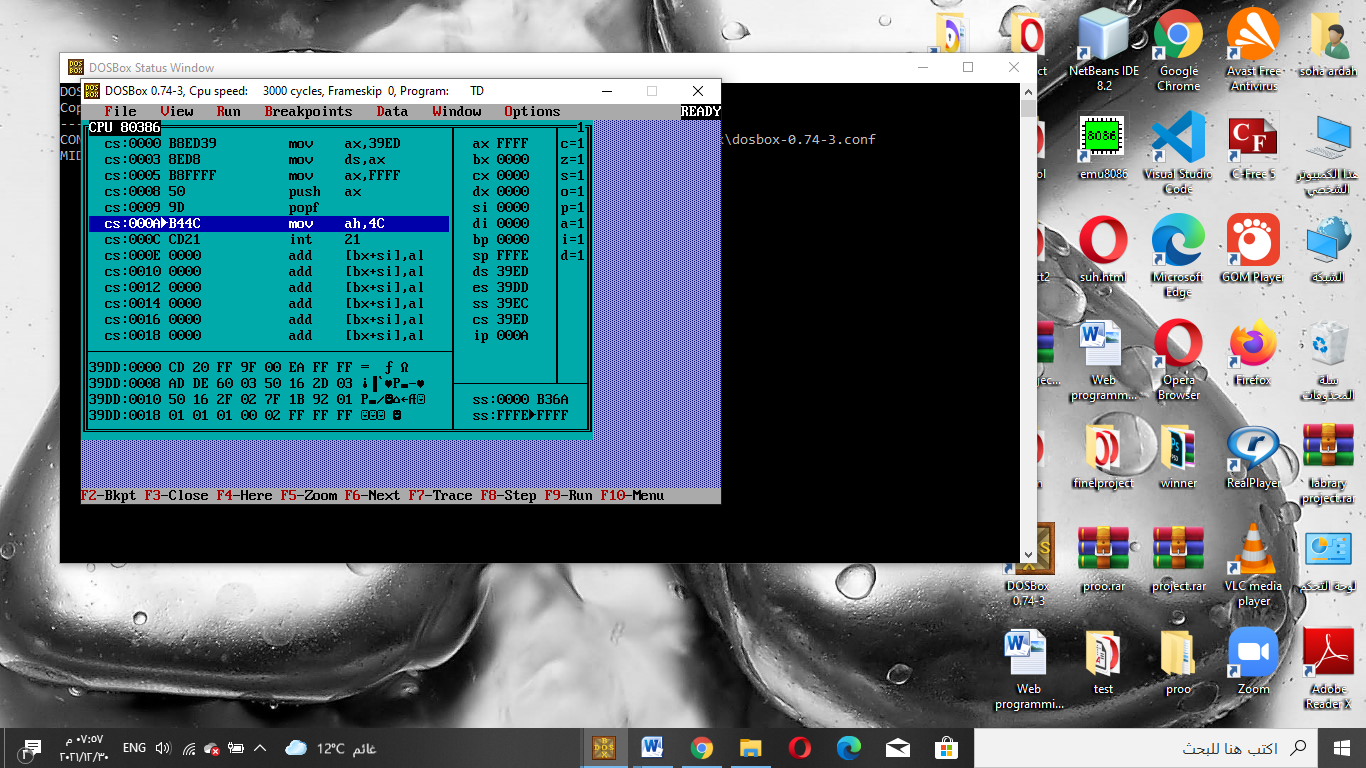
popf

mov ah,4ch

int 21h

end main

end



Tasks

2) Apply password task.

3) and write a subroutine that accept two 8bit number and determine if they have common divisor or not then.

4) combine 2+3

5) make the program 4 operate continuously if enter any number "q" stop the program.

dosseg

.model small

.data

pwd db "assembly "

msg1 db 10,13," the two number is diviser " ,10,13 ,"$"

msg2 db 10,13," the two number is not diviser " ,10,13 ,"$"

msg3 db 10,13," the pass is correct emter thje number plz",10,13,"$"

msg4 db 10,13," uncorrect pass",10,13,"$"

.code

main:

mov ax,@data

mov ds,ax

mov bx,offset pwd

mov cx,8

x: mov ah,8

int 21h

push ax

push [bx]

inc bx

mov ah,2

mov dl,"\*"

int 21h

loop x

mov cx,8

y:

pop ax

pop bx

cmp al,bl

jne error

loop y

mov ah,9

mov dx,offset msg3

int 21h

mov ah,1

int 21h

cmp al,"q"

je exit

sub al,30h

mov bl,al

mov ah,1

int 21h

cmp al,"q"

je exit

sub al,30h

mov bh,al

mov al,bl

div bh

cmp ah,00h

je rt

er: mov ah,9

mov dx,offset msg2

int 21h

jmp exit

rt:mov ah,9

mov dx,offset msg1

int 21h

exit:

mov ah,4ch

int 21h

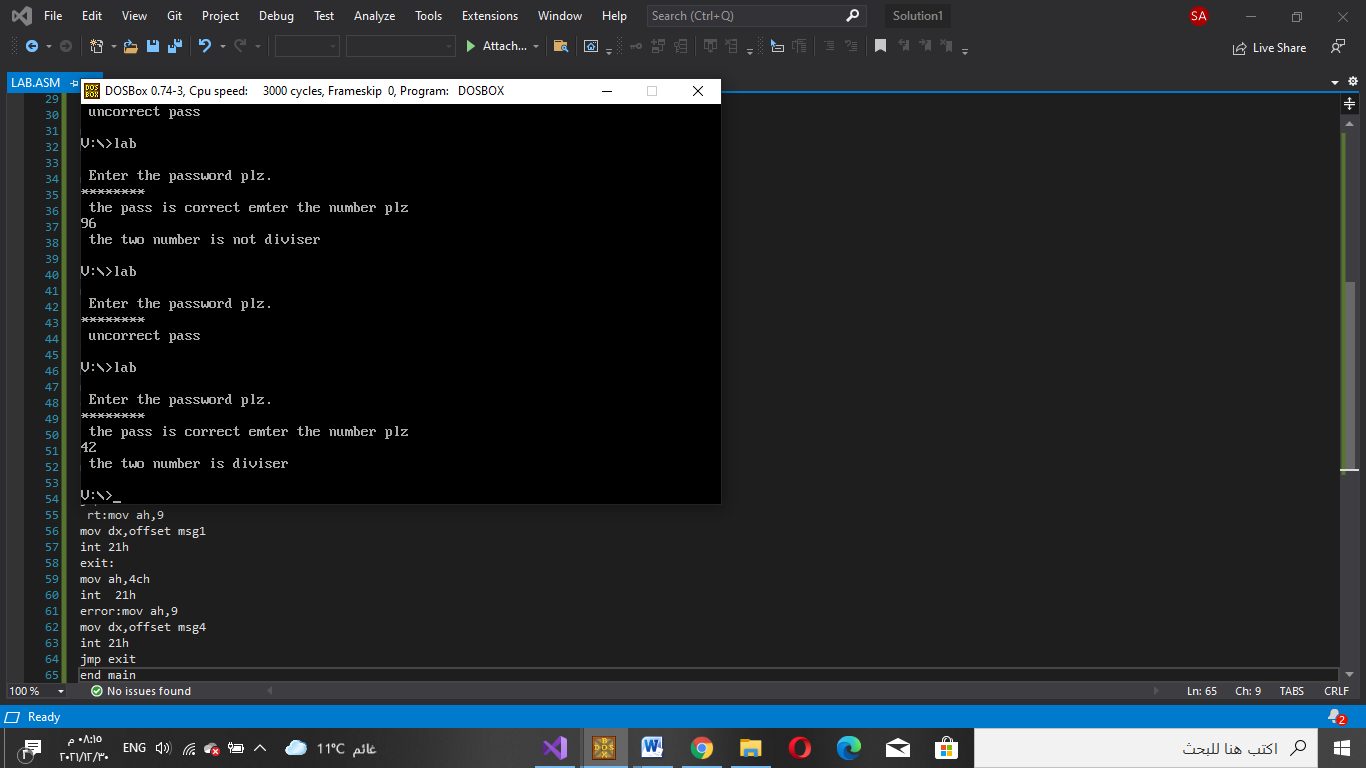
error:mov ah,9

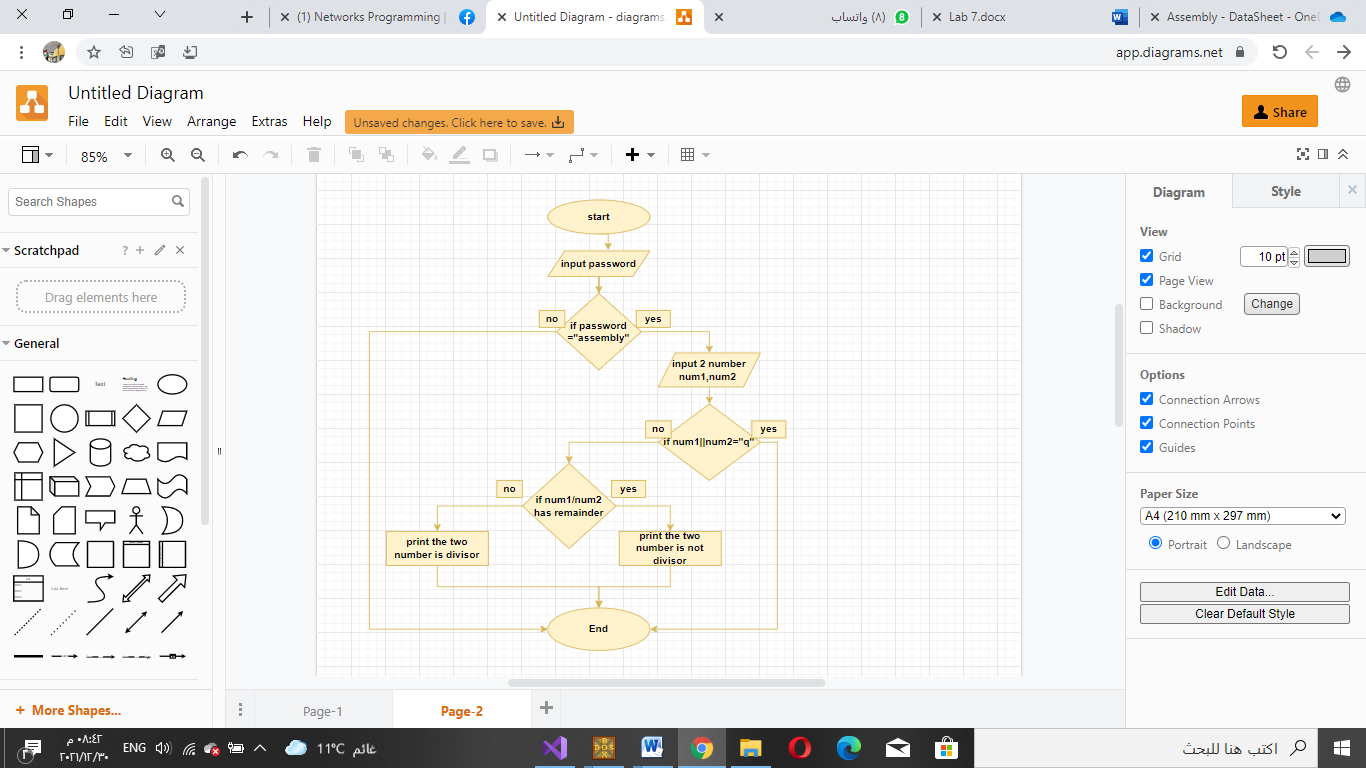
mov dx,offset msg4

int 21h

jmp exit

end main





**Conclusion**:

In this experiments we know many thing:

Large programs are hard to handle :

We can break them to smaller programs They are called subroutines .

Subroutines are called from the main program Writing subroutines

we should jump use CALL)

we return to(use RETURN)

The word stack is used because storage/retrieval of words in the stack memory area is the same as accessing items from a stack of items.