CENG 222 Computer Organization

1)Analyze the assembly code below:

```
.data
msg db "AB",0dh,0ah,'$'

.code
main proc
        mov ax,@data
        mov ds,ax

; Add your code here
; ------------------

; ------------------

        mov ah,9
        mov dx,offset msg
        int 21h

        mov ax, 4c00h
        int 21h
main endp

end main
```

The program above prints "AB" to the console. The message "AB" is stored at msg in data segment. Therefore, "offset msg" is the starting address of the string. Modify the code according to the following experiments:

1- Copy the second byte stored at **"msg" (which is 'B' currently)** to the first byte. The output should be "BB". In order to do that, load 2-bytes data stored at **"msg"** into a **16-bits** register. Copy one part of the register to the other part. Then, write your register back to msg.
2- Copy the first byte stored at **"msg" (which is 'A' currently)** to the second byte. The output should be "AA". To achieve this, load the first byte stored at **"msg"** into one **8-bits** register (high or low part) and copy it onto the other part and write the data stored in register back to msg.
3- Load the data stored at **"msg"** into one register and swap the high and low parts of the register using a temporary **8-bits** register. Then write back your "swapped" register onto msg. The output should be "BA".

2)

Write the following C code in assembly using no jump instructions (use loop command instead) You should not use other registers to store the value of CX temporarily. You should use the stack properly to store the value of CX properly.

Hint: Don't forget that loop command only operates on CX and to use a inner loop you must remember the value of CX for the outer loop at each iteration.

```
int b =0;
for(int i=0; i<5;i++)
        for(int j=0; j<10;j++)

                b = I + 2*j;
```