

CEN 214 Microprocessors Lab Assignment 11

New Instruction Set Commands:

CALL targetOperand (PROC-ENDP, RET)

Saves procedure linking information on the stack and branches to the called procedure specified using the target operand. The target operand specifies the address of the first instruction in the called procedure. The operand can be an immediate value, a general-purpose register, or a memory location.

Example:

```
CALL p1
ADD AX, 1
RET          ; return to OS.

p1 PROC      ; procedure declaration.
    MOV AX, 1234h
    RET      ; return to caller.
p1 ENDP
```

Examples

```
MOV AX, 0100h
MOV DS, AX

JMP DEVAM
k1 DB 'Hello ',10,13,0
k2 DB 'World!',10,13,0
DEVAM:

LEA BX, k1
CALL Yaz
LEA BX, k2
CALL Yaz

MOV AH, 4Ch
INT 21h

;Procedure Yazdir
Yaz PROC
    MOV AH, 0Eh
DONGU:
    MOV AL, [BX]
    CMP AL, 0
    JE SON
    INT 10h
    INC BX
    JMP DONGU
SON:
    RET
Yaz ENDP
```

The assembly code above demonstrate a simple program that prints out any variable that is loaded to BX with LEA and CALL Yaz procedure. But procedure Yaz is not register and flag safe for every condition. Append necessary code to the Yaz procedure to make sure it is safe for every condition.(Use Push(a/f)/Pop(a/f) instructions.)

1. Use the code above and write an uppercase print procedure (prints out every character in uppercase) named BuyukYaz. (Hint: look at assignment 7 for code.)
2. Use the code above and write a lowercase print procedure (prints out every character in lowercase) named KucukYaz. (Hint: look at assignment 7 for code.)