Universidad Nacional de Luján
41406 - Arquitectura de Computadoras
2º Parcial – 13/11/2017

1. Realizar un programa en assembler que pida el ingreso de 20 caracteres por teclado. Los mismos deben almacenarse en un arreglo llamado cadena. Luego deberán mostrarse por pantalla los caracteres que sean mayores que el promedio de todos los ingresados y tengan el bit 0 y bit 1 distintos. Utilizar procedimientos.
   Una idea: dividir la funcionalidad en el ingreso de los caracteres por teclado, el cálculo del promedio y el recorrido del arreglo para mostrar los caracteres adecuados.

```
.model small

.data
  cadena db 20 dup (?)

.code
  mov ax, @data
  mov ds, ax
  call ingreso
  call promedio
  call mostrar
  mov ah, 4ch
  int 21h

  ingreso proc
    mov ah, 1
    xor bx, bx
    xor dx, dx
    mov cx, 20
    mov si, 0
repetir:
    int 21h
    mov cadena[si], al
    mov dl, al
    add bx, dx
    inc si
   loop repetir

    ret
ingreso endp

promedio proc
    mov ax, bx
    mov cl, 20
    div cl
    ret
promedio endp
```

```
mostrar proc
        mov ah, 2
        mov si, 0
    procesar:
        mov dl, cadena[si]
        cmp dl, al
        jbe seguir
        mov dh, dl
        and dh, 3
        cmp dh, 0
        je seguir
        cmp dh, 3
        je seguir
        int 21h
    seguir:
        inc si
    loop procesar
    ret
mostrar endp
```

2. Realizar un procedimiento en assembler que recorra un área de memoria cuyo comienzo es pasado como parámetro en la pila (stack) y que finaliza en la primer posición (posterior a esa) que tenga almacenado el valor FFh. A los valores recorridos se los transformará de la siguiente manera:
   $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 \longrightarrow b_3 b_2 b_1 b_0\ 0\ 0\ b_5 b_4$ y luego volverá a almacenar en la posición de memoria original. No es necesario hacer el programa principal
   Una idea: realizar la transformación del valor en un procedimiento aparte.

```
Transformar proc
        pop bx
        pop si
        push bx
recorrer:
        mov al,[si]
        cmp al, 0ffh
        jz finalizar
        shl al, 2
        rol al, 2
        mov [si], al
        inc si
        jmp recorrer

finalizar:
        ret
Transformar endp
```