Enunciado MUE

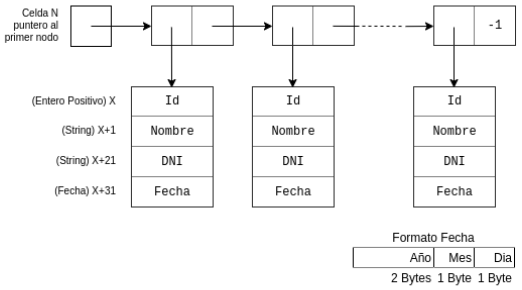
La siguiente estructura de datos es una lista de datos de personas. Los datos son Nombre, número de DNI y fecha de nacimiento.

Cada nodo de la lista contiene un puntero a un registro de datos y un puntero al siguiente nodo.

El registro consta de 32 celdas donde la primer celda contiene un Id (entero positivo), las siguientes 20 celdas un String donde se guarda el nombre, las siguientes 10 celdas el número de DNI de la persona, y la última celda la fecha.

El formato fecha es una representación de la fecha en una celda, en la cual se utilizan los 2 bytes más significativos para el año, seguido de 1 byte para el mes y el byte menos significativo para el día.

La celda N contiene el puntero al primer nodo.



Escribir el código de la subrutina CUMPLEN que recibe como parámetro una fecha y la dirección de memoria N con el puntero a la estructura de datos. La subrutina debe **buscar todas las personas que cumplen años ese día e imprimir por pantalla la lista** de DNI y Nombre. Utilizar la subrutina PRINT que recibe 2 parámetros cada uno con la dirección de memoria de un string y escribe en pantalla formateado.

NOTA: Todas las direcciones de memoria están en el Data Segment. Cuando se dice puntero se entiende como una dirección de memoria donde se encuentra una celda con un dato o es la primera de una estructura. **Debe especificar el formato de invocación** de ambas subrutinas y **desarrollar solo la solicitada**.

```
empty Equ -1
last Equ  -1
; invocación subrutina cumplen
push <fecha>
push <dirección celda N>
call cumplen
add sp, 2
stop


cumplen:          push bp
                  mov bp,sp
                  push ax
                  push bx
                  push cx
                  push dx

                  mov ax, [bp+2]     ; dirección celda N
                  cmp [ax], empty
                  jmp finCumplen     ; no había lista
                  mov ax, [ax] ; dirección nodo actual
loop:             mov bx, [ax] ; pos 0 del registro
                  mov cx, [bx+31]  ; fecha nacimiento persona
                  mov dx, [bp+3]     ; fecha pasada por parámetro
                  and cx, %FFFF      ; me quedo con mes y dia
                  and dx, %FFFF      ; me quedo con mes y dia
                  cmp cx, dx
                  jnz buscaOtro
                  ; cumple, llamo a Print para mostrarlo por pantalla
                  push [bp+21] ; dni
                  add bx, 1
                  push bx            ; paso dirección inicial nombre
                  call print
                  add sp, 2
                  jmp buscaOtro

buscaOtro:        mov ax, [ax+1]     ; paso al siguiente nodo
                  cmp ax, last
                  jnz loop


finCumplen:       pop dx
                  pop cx
                  pop bx
                  pop ax
                  mov sp, bp
                  pop bp
                  ret
```

Enunciado MUQ

Buscar: MUQ

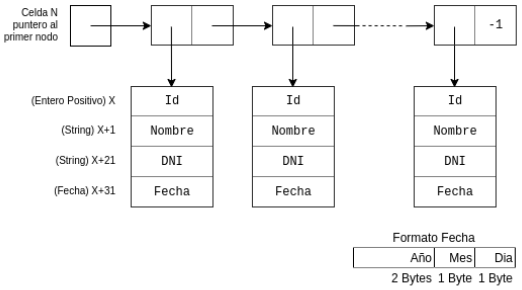
La siguiente estructura de datos es una lista de datos de personas. Los datos son Nombre, número de DNI y fecha de nacimiento.

Cada nodo de la lista contiene un puntero a un registro de datos y un puntero al siguiente nodo.

El registro consta de 32 celdas donde la primer celda contiene un Id (entero positivo), las siguientes 20 celdas un String donde se guarda el nombre, las siguientes 10 celdas el número de DNI de la persona, y la última celda la fecha.

El formato fecha es una representación de la fecha en una celda, en la cual se utilizan los 2 bytes más significativos para el año, seguido de 1 byte para el mes y el byte menos significativo para el día.

La celda N contiene el puntero al primer nodo.



Escribir el código de una subrutina INSERT que recibe como parámetros una la dirección de memoria N con el puntero a la estructura de datos, un puntero al string de un nombre, un puntero al string de un DNI y una fecha. La subrutina debe **crear e insertar en un nodo en la lista ordenada por Nombre**. Utilizar la subrutina NEW, que devuelve en AX el puntero a un nodo creado, pasándole por parámetro un puntero al nombre, puntero al DNI y fecha.

NOTA: Todas las direcciones de memoria están en el Data Segment. Cuando se dice puntero se entiende como una dirección de memoria donde se encuentra una celda con un dato o es la primera de una estructura. **Debe especificar el formato de invocación** de ambas subrutinas y **desarrollar solo la solicitada**.

```
; llamado a subrutina insert
push <puntero celda N>     ; bp+5
push <puntero a nombre>     ; bp+4
push <puntero a dni>        ; bp+3
push <fecha Nacimiento>    ; bp+2
call INSERT
add sp, 4
stop

insert:                 push bp
                        mov bp, sp
                        push ax
                        push bx
                        push cx
                        push dx
                        push ex
                        push fx
                        push ac

                        mov ax, [bp+5]     ; dirección celda N
                        cmp [ax], -1 ; apunta a algo ??
                        jz fin
                        mov fx, [bp+4]     ; puntero a nombre a insertar
                        mov ex, -1         ; en ex guardo posición nodo anterior
                        mov ax, [ax] ; me paro en primer nodo
loop:                   mov bx, [ax] ; obtengo posición cero del registro
                        mov dx, bx         ;
                        add dx, 1          ; pos inicial nombre a comparar
                        scmp dx,fx
                        jp inserta         ; inserta cuando es menor al actual
```

```
                        ;si no,avanza
                        mov ex, ax
                        mov ax, [ax+1]
                        cmp ax, -1
                        jz inserta        ; inserto al final
                        jmp loop
;necesito reservar memoria 2 veces
; una para el nodo
; otra para el arreglo
inserta:
            mov cx, 2
            sys 5
            ; en rigor, debería verificar si hay espacio, pero son solo 2 nodos
            cmp ex, -1  ;si anterior es -1, modifico cabeza
            jz cabeza
            ; engancho nodo
            mov [ex+1], dx
            mov [dx+1], ax

insertaReg: push [bp+4]
            push [bp+3]
            push [bp+2]
            call new    ; retorna en Ax, dirección del registro creado
            add sp, 3
            mov [dx], ax; engancha registro al nuevo nodo

fin:        pop ac
            pop fx
            pop ex
            pop dx
            pop cx
            pop bx
            pop ax


cabeza:     mov ac, [bp+5]    ; actualizo celda M con nueva cabeza
            mov [ac], dx
            mov [dx+1], ax
            jmp insertaReg
```