

## ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES II (2º)

Ingeniería Informática – Escuela Politécnica Superior – UAM

Parcial - Curso 08-09

NOMBRE : \_\_\_\_\_ DNI : \_\_\_\_\_  
APELLIDOS : \_\_\_\_\_

P1. Suponiendo que **CS=1000h**, **DS=2000h**, **ES=4321h**, **SS=1111h**, **BX=2222h**, **BP=3333h** y **SI=0002h**, indicar la **dirección física** de memoria (@) a la que se está accediendo en cada una de las siguientes instrucciones, considerando los registros de segmento por defecto. (1 punto)

mov AH, 4[BX][SI]	@ = 22228h
mov AH, SS:[BP][SI]	@ = 14445h
mov AL, [BP + 4]	@ = 14447h
mov AL, CS:[1000h]	@ = 11000h

P2. Suponiendo que **CS=2000h**, **DS=204Fh**, **ES=204Dh**, **SS=2222h**, **BX=0020h** y **SI=0002h**, Indicar el valor del **registro AX** tras ejecutar cada una de las instrucciones siguientes (**independientes entre si**), dado el volcado de memoria adjunto. Expresar los **dígitos hexadecimales desconocidos de AX con un '?'**. (1 punto)

204F:0000 73 65 67 20 00 68 61 6E

mov AX, [SI]	AX = 2067h
mov AH, 3[SI]	AX = 68??h
mov AL, ES: [BX + 5]	AX = ??68h
mov AX, ES: [20h]	AX = 6573h

P3. Si **SP=0006h** y **FLAGS=0210h** al inicio de la ejecución del código que se adjunta, indicar los valores contenidos en las **primeras seis posiciones de la pila** al ejecutar la primera instrucción del procedimiento **Leer\_Datos**, tanto cuando todos los procedimientos del programa son cercanos (**NEAR**), como cuando son lejanos (**FAR**). La pila está inicializada a ceros. (1 punto)

```
2100:2250 E8A8FD    call Leer_Datos
2100:2253 89161000    mov Datos[0], dx
```

0	1	2	3	4	5	0	1	2	3	4	5
0	0	0	0	53h	22h	0	0	53h	22h	0	21h
Caso NEAR						Caso FAR					

**P4.** Indicar el vector de la interrupción de **punto de ruptura** (*breakpoint*) dado el siguiente volcado de memoria. (1 punto)

```
0000:0000 54 02 CF 15 CE 01 CF 15 04 00 70 00 D7 01 CF 15
0000:0010 04 00 70 00 30 00 00 C8 30 00 00 C8 30 00 00 C8
```

Segmento = 15CFh      Offset = 01D7h

**P5.** Se tiene una matriz bidimensional de tamaño (**FILAS** x **COLUMNAS**) almacenada por filas en la variable **Matriz2D**. Escribir en ensamblador un procedimiento lejano, **escribelcol**, que reciba la **dirección de la matriz en el registro BX** y ponga a **uno** todos los elementos de la **columna indicada en el registro AX**. Se valorará la eficiencia del código. (2 puntos)

```
FILAS = 10
COLUMNAS = 20
Matriz2D db FILAS*COLUMNAS dup (?)

mov bx, offset Matriz2D
mov ax, 4
call escribelcol      ; Pone a 1 los elementos de la columna 4
                      ; de Matriz2D

escribelcol PROC FAR

    push cx, si

    mov cx, FILAS      ; Itera el número de filas dado
    mov si, ax          ; Índice a primer elemento de columna dada

buclecol:  mov BYTE PTR [bx][si], 1
           add si, COLUMNAS      ; Índice pasa a siguiente fila
           dec cx
           jnz buclecol

    pop si, cx
    ret

escribelcol ENDP
```

**P6.** Usando los procedimientos lejanos **enviar0** y **enviar1**, escribir en ensamblador un procedimiento eficiente que envíe secuencialmente los bits del **registro AL**, desde el más significativo al menos significativo. Se valorará la eficiencia del código. (2 puntos)

```
enviarAL PROC

    push cx

    mov cx, 8          ; Itera los ocho bits de AL

bucle:    rcl al, 1      ; Pasa el bit más alto de AL al acarreo
           jc envia1     ; Si hay acarreo envía 1, si no envía 0

           call enviar0

           jmp finbucle

enviarAL ENDP
```

```

envial:    call enviar1

finbucle:  dec cx
           jnz bucle

           rcl al, 1           ; Deja AL igual que al principio

           pop cx
           ret

enviarAL ENDP

```

**P7.** Escribir en ensamblador el código necesario para **poner a 1 los bits 5, 10 y 14** del **registro AX**, dejando todos los demás bits de ese registro intactos, y **poner a 0 los bits 5, 10 y 14** del **registro BX**, dejando intactos los demás bits. Se valorará la eficiencia del código. (2 puntos)

```

or ax, 0100010000100000b      ; 4420h
and bx, 1011101111011111b     ; BBDfh

```