

SISTEMAS BASADOS EN MICROPROCESADORES
Grado en Ingeniería Informática
Escuela Politécnica Superior – UAM

Parcial 1 - Modelo A
Curso 2016/17

NOMBRE : _____ **SOLUCION** _____ **DNI :** _____
APELLIDOS : _____

P1. Suponiendo que **CS=0000h, DS=1000h, ES=FFFFh, SS=2000h, BX=2222h, BP=0000h** 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. (1,5 puntos)

<code>mov AH, ES:16[SI]</code>	@ = 00002h (0,3)
<code>mov AH, 16[SI]</code>	@ = 10012h (0,3)
<code>mov AL, [BP - 2]</code>	@ = 2FFFEh (0,3)
<code>mov AL, CS:[FFFFh]</code>	@ = 0FFFFh (0,3)
<code>mov AL, DS:[BP - 1]</code>	@ = 1FFFFh (0,3)

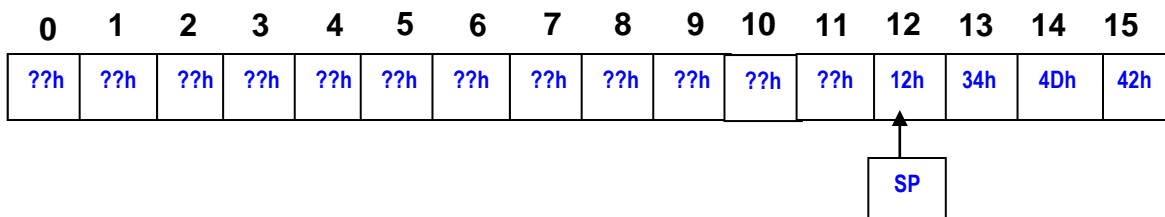
P2. Suponiendo que **CS=2000h, DS=204Fh, ES=204Fh, SS=2000h, BX=0001h, BP=04F8h, DI=0007h** y **SP=04F8h**, Indicar el valor del **registro AX** tras ejecutar cada una de las instrucciones siguientes (**independientes entre sí**), dado el volcado de memoria adjunto. Expresar los **dígitos hexadecimales desconocidos de AX** con un '?'. (1,5 puntos)

204F:0000 73 65 67 20 00 68 61 6E
 204F:0008 12 34 4E 00 FF 00 33 11

<code>mov AH, [BX][DI]</code>	AX = 12??h (0,3)
<code>mov AL, 3[DI]</code>	AX = ??4Eh (0,3)
<code>mov AX, [BP - 6]</code>	AX = 2067h (0,3)
<code>pop AX</code>	AX = 3412h (0,3)
<code>mov AX, 16[BX]</code>	AX = ???h (0,3)

P3. Suponiendo que **SS=424Dh, SP=16, AX=3412h** y **BX=5678h**, indicar el **valor hexadecimal** de los **16 primeros bytes** del **segmento SS** una vez ejecutado el siguiente programa. (2 punto)

```
push AX
pop BX
push SS
push BX
```



P4. Declarar mediante directivas de ensamblador de 8086 las mismas variables que aparecen en el siguiente extracto en lenguaje C, teniendo en cuenta que las cadenas de caracteres en C acaban con el byte 0. (2 puntos)

```
char nombre[20];           // Cadena de caracteres de 20 bytes
short edad = 17;          // Entero de 2 bytes inicializado
char tabla2D[10][2];      // Tabla de bytes de 10 filas por 2 columnas
short valores[5] = { 1, 2, 3, 4, 5 };
char despedida[20] = "Hasta luego";
```

```
nombre    db 20 dup(?)      (0,4)
edad      dw 17             (0,4)
tabla2D   db 20 dup(?)      (0,4)
valores   dw 1,2,3,4,5      (0,4)
despedida db "Hasta luego",0 (0,4)
          db 8 dup(?)        (0,4)
```

P5. El siguiente programa en lenguaje ensamblador de 8086, que debe **invertir el orden de los caracteres de una cadena dada de 512 bytes como máximo**, tiene varios errores. Proponer una versión correcta del mismo programa haciendo el **menor número de cambios**. Sólo es necesario reescribir las líneas erróneas. (3 puntos)

```
datos segment
    cadena    dw "Hola"
    longitud  db  cadena-$
datos ends

resultados segment
    resultado db 200 dup (?)
resultados ends

codigo segment
    assume cs:codigo, ds:datos
    invertir proc far
        mov ax, datos
        mov ds, ax
        mov ax, resultado
        mov es, ax
        mov si, longitud
        mov di, 0
seguir:  mov al, cadena[si-1]
        mov resultado[di], al
        dec si
        inc di
        jz seguir
        mov ax, 4C00h
        int 21h
    invertir endp
codigo ends
end codigo
```

```
cadena      db "Hola"          (0,4)
longitud    dw  $-cadena        (0,4)

resultado   db 512 dup (?)      (0,4)

assume cs:codigo, ds:datos, es:resultados (0,1)

mov ax, resultados                (0,4)

mov es:resultado[di], al          (0,4)
inc di                            (0,2)
dec si                            (0,2)
jnz seguir                        (0,4)

end invertir                      (0.1)
```