

ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES II (2º)

Ingeniería Informática – Escuela Politécnica Superior – UAM

Final Teoría - Curso 2010-11

NOMBRE : _____ DNI : _____
 APELLIDOS : _____

P1. Suponiendo que **CS=2000h**, **DS=424Eh**, **ES=424Dh**, **SS=424Eh**, **BX=0004h**, **BP=0003h** y **DI=FFFFh**, Indicar el valor del **registro AX** tras ejecutar cada una de las instrucciones siguientes (**independientes entre ellas**), dado el volcado de memoria adjunto. Expresar los **dígitos hexadecimales desconocidos de AX con un '?'**. (1,5 puntos)

424E:0000 4E 42 74 61 20 65 73 20

mov AX, [BX][DI]	AX = 2061h
mov AX, [0000h]	AX = 424Eh
mov AL, DS:[BP]	AX = ??61h
mov AH, [BP]13	AX = ????h
mov AX, 2[DI]	AX = 7442h

P2. La función de lenguaje C cuya signatura se indica en el recuadro de la derecha es invocada desde el programa de código máquina que se muestra en el recuadro de la izquierda. En el momento anterior de la llamada, se suponen los siguientes valores del puntero de pila y de los parámetros de la función: **SP = 16**, **n = 4321h**, **c = 12h**, **p = 1234h:8765h**.

5342:FF0A E8F6FF call _fun 5342:FF0D B8004C mov ax, 4C00h	fun (char* p, int n, char c);
--	---------------------------------

Indicar el valor de las 16 posiciones iniciales de la pila en el momento de ejecutarse la primera instrucción de código máquina de la función `fun`, tanto cuando todas las direcciones son cercanas (**NEAR**), como cuando son lejanas (**FAR**). (3 puntos)

Caso NEAR:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
								0Dh	FFh	65h	87h	21h	43h	12h	00h

Caso FAR:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				0Dh	FFh	42h	53h	65h	87h	34h	12h	21h	43h	12h	00h

P3. Indicar el vector de la interrupción de **impresión por pantalla** 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 = **C800h** Offset = **0030h**

P4. Escribir en ensamblador el código necesario para **poner a 1 los bits 0, 7 y 14 del registro AX**, dejando todos los demás bits de ese registro intactos, y **poner a 0 los bits 2, 10 y 15 del registro BX**, dejando intactos los demás bits. Se valorará la eficiencia del código. (1,5 puntos)

```
or ax, 0100000010000001b                   ; 4081h
and bx, 0111101111111011b                 ; 7BFBh
```

P5. Escribir en ensamblador utilizando instrucciones básicas (sin instrucciones de manipulación de cadenas ni de bucles) la función `strlen` de C, cuya signatura se reproduce a continuación. Esta función retorna la longitud de la cadena de caracteres que recibe como argumento. Dicha cadena acaba con un byte a cero. Se supone que el programa de C está compilado en **modelo largo**. Se valorará la eficiencia del código. (3 puntos)

```
int strlen (char *s);

_strlen PROC far
    push bp
    mov bp, sp

    push ds bx

    lds bx, 6[bp]   ; Lee offset y segmento
    mov ax, 0       ; Inicializa contador

itera:  cmp BYTE PTR [bx], 0   ; Test de final de cadena
        je fin
        inc ax                 ; Incrementa contador de caracteres
        inc bx                 ; Apunta al siguiente carácter
        jmp itera

fin:    pop bx ds
        pop bp

        ret

_strlen ENDP
```