



Apellidos, Nombre:

Gestión

Sistemas

Test (3 puntos)

Pregunta correcta = 0,3 Pregunta no contestada = 0 Pregunta incorrecta = - 0,1

1. Un programa ensamblador:
 - Se corresponde con un programa compilado de alto nivel.
 - Traduce a lenguaje máquina un programa de bajo nivel.
 - Traduce a fichero ejecutable un programa de bajo nivel.
 - Es un sistema operativo para algunas arquitecturas de ordenador.
2. ¿Cual de las siguientes instrucciones no modifican necesariamente la secuencia normal de ejecución de un programa?
 - JMP dir1
 - JNE dir1
 - CALL dir1
 - RET
3. ¿Qué registros del microprocesador 8086 se pueden utilizar para realizar direccionamientos relativos?
 - AX, BX, CX, y DX
 - SI, DI y AX
 - BX, BP y DX
 - BX, BP, SI y DI
4. Señale la opción **correcta**:
 - No es posible acceder a una sesión MS-DOS en Windows 2000.
 - El sistema operativo Windows 2000 permite iniciar el sistema en modo MSDOS.
 - El sistema operativo Windows 2000 es un sistema operativo multiusuario.
 - El sistema operativo MS-DOS es un sistema operativo multiusuario.
5. Señale la opción **correcta**:
 - En Windows 2000 todos los usuarios tiene los mismos privilegios y permisos.
 - En Windows 2000 todos los usuarios tienen diferentes privilegios y permisos.
 - En Windows 2000 los diferentes usuarios pueden tener diferentes permisos y privilegios.
 - En Windows 2000 no se pueden asignar permisos ni privilegios a los usuarios.
6. ¿Cuál de los siguientes nombres de fichero no está permitido en MS-DOS?
 - Ficheros.txt
 - Libros.exe
 - 2archivo.doc
 - misdocumentos.asm



Examen de Laboratorio de Estructura de Computadores
I.T. Informática de Gestión / Sistemas. 22 de enero de 2009
Departamento de Automática - ATC -

Apellidos, Nombre:

Gestión

Sistemas

7. En MS-DOS, la orden **dir *.doc** muestra
- todo el contenido del directorio actual.
 - todos los ficheros de texto del directorio actual.
 - todos los ficheros con la extensión doc del directorio actual.
 - el fichero de nombre *.doc.
8. ¿Cual de las siguientes **no** es una función de un sistema operativo?
- Planificar recursos.
 - Permitir compartir dispositivos.
 - Decodificar las instrucciones.
 - Facilitar la entrada-salida.
9. La memoria RAM:
- Contiene las rutinas de la BIOS del sistema.
 - Salvaguarda el registro de estado y el contador de programa en caso de fallo eléctrico.
 - Se accede a su contenido direccionando una posición por el bus de datos.
 - Ninguna de las anteriores.
10. Para modificar el contenido del registro AX dentro de CODEVIEW se debe indicar en la línea de comandos la orden:
- EB AX nuevo_contenido.
 - EW AX nuevo_contenido.
 - WW AX nuevo_contenido.
 - R AX nuevo_contenido.



Apellidos, Nombre:

Gestión

Sistemas

Programación en Ensamblador (7 puntos)

1) Para el programa representado por el código fuente siguiente:

<pre>DOSSEG .MODEL SMALL .STACK 100h .DATA Dato1 DW 4234h,0F5A1h Dato2 DW 5234h,5678h Dato3 DW 4567h Dato4 DW 1245h ResulResta DW ?,? ResulMult DW ?,?</pre>	<pre>.CODE mov ax,@DATA mov ds,ax ;Resta (apartado a) ;Multiplicación (apartado b) mov ah,4Ch int 21h END</pre>
--	---

a) Escriba el trozo de código necesario para restar **Dato2** de **Dato1** y almacenar el resultado en **ResulResta**.

SOLUCIÓN

```
mov ax,Dato1
sub ax,Dato2
mov ResulResta,ax
mov ax,Dato1+2
sbb ax,Dato2+2
mov ResulResta+2,ax
```

b) Escriba el trozo de código necesario para multiplicar **Dato3** por **Dato4** y almacenar el resultado en **ResulMult**.

SOLUCIÓN

```
mov ax,Dato3
mul Dato4
mov ResulMult,ax
mov ResulMult+2,dx
```



Examen de Laboratorio de Estructura de Computadores
I.T. Informática de Gestión / Sistemas. 22 de enero de 2009
Departamento de Automática - ATC -

Apellidos, Nombre:

Gestión

Sistemas

- 2) En el código fuente mostrado a continuación de este párrafo, el dato **Tabla** es una secuencia de códigos ASCII que representa a tres números decimales de dos dígitos cada uno. Escribe el trozo de código necesario para agrupar los dos dígitos de cada número y formar su valor en BCD empaquetado. A continuación lo debe guardar en la posición correspondiente de la tabla **Resultado**. Por ejemplo, los dos primeros códigos ASCII, 34h y 35h, corresponden a los dígitos 4 y 5, una vez agrupados formarían el número 45 en BCD empaquetado que debe almacenarse en la primera posición de **Resultado**.

<pre>DOSSEG .MODEL SMALL .STACK 100h .DATA Tabla DB 34h, 35h, 36h, 37h, 38h, 39h Resultado DB 3 DUP(?)</pre>	<pre>.CODE mov ax,@DATA mov ds,ax ;Trozo de código pedido mov ah,4Ch int 21h END</pre>
---	--

SOLUCIÓN

```
mov si, OFFSET Tabla
mov di, OFFSET Resultado
mov bx,3
mov cl,4
Lazo:
    mov al,[si]
    sub al,30h
    shl al,cl
    inc si
    mov dl,[si]
    sub dl,30h
    add al,dl
    inc si
    mov [di],al
    inc di
    dec bx
    jnz Lazo
```

OTRA SOLUCIÓN

```
mov si, OFFSET Tabla
mov di, OFFSET Resultado
mov bl,3
Lazo1:
    mov cx,2
    xor al,al
    mov bh,16
    Lazo2:
        mul bh
        mov dl,[si]
        sub dl,30h
        add al,dl
        inc si
        loop Lazo2
    mov [di],al
    inc di
    dec bl
    jnz Lazo1
```



Apellidos, Nombre:

Gestión

Sistemas

- 3) El código fuente mostrado a continuación de este párrafo es de un programa que convierte una letra leída del teclado en mayúsculas o en minúsculas según su tipo original. El programa solicita una letra y a continuación la muestra convertida en mayúscula / minúscula, reiterando el proceso hasta que se pulsa la tecla <enter> como respuesta a la petición de letra. El programa trabaja con el alfabeto sin ñ.

Código fuente del programa propuesto	
DOSSEG	
.MODEL SMALL	
.STACK 100h	
.DATA	
	Msj1 DB 10,13,'Introduzca una letra: \$'
	Msj2 DB 10,13,'La letra convertida es: \$'
.CODE	
Inicio:	mov ax,@DATA mov ds,ax
Otra:	PedirLetra cmp al,13 je Fin Call ConvMayMin ImprimeResul jmp Otra
Fin:	mov ah,4Ch int 21h END Inicio

Ejemplo de la información mostrada en la pantalla durante una ejecución del programa:

```

Introduzca una letra: a
La letra convertida es: A
Introduzca una letra: b
La letra convertida es: B
Introduzca una letra: Z
La letra convertida es: z
Introduzca una letra: 2
La letra convertida es: 2
Introduzca una letra: <enter>
C:>
  
```

Tabla de códigos ASCII					
Decimal	Hexadecimal	Símbolo	Decimal	Hexadecimal	Símbolo
10	0A	Sato de línea	65	41	A
13	0D	Ret. de carro <enter>	66	42	B
	
48	30	0	90	5A	Z
49	31	1		...	
	...		97	61	a
57	39	9	98	62	b
				...	
			122	7A	z



Apellidos, Nombre:

Gestión

Sistemas

- a) Escriba la macro **PedirLetra** que muestra el mensaje de petición de la letra, la lee del teclado y la devuelve en el registro AL.

SOLUCIÓN

```
PedirLetra    MACRO
              mov dx,OFFSET Msj1    ;Imprimir mensaje
              mov ah,9
              int 21h
              mov ah,1              ;Leer tecla
              int 21h
              ENDM
```

- b) Escriba la macro **ImprimeResul** que imprime el mensaje de presentación del resultado y a continuación la letra convertida.

SOLUCIÓN

```
ImprimeResul  MACRO
              mov bl,al ;Guardamos el ASCII a imprimir en BL
              ;porque 09h - Int 21h devuelve $ en
              ;AL
              mov dx,OFFSET Msj2    ;Imprimir mensaje
              mov ah,9
              int 21h
              mov dl,bl
              mov ah,2              ;Imprimir letra
              int 21h
              ENDM
```



Apellidos, Nombre:

Gestión

Sistemas

- c) Escriba el procedimiento **ConvMayMin** de conversión mayúsculas / minúsculas al que se le pasa la letra en el registro AL y la devuelve en el mismo registro convertida a mayúscula / minúscula según su tipo original.

SOLUCIÓN

```
ConvMayMin PROC
    cmp al, 61h ;61h = ASCII letra a
    jb NoMinus
    cmp al, 7Ah ;7Ah = ASCII letra z
    ja NoMinus
    and al, 11011111b ;Convierte 6 ó 7 en 4 ó 5
    jmp Imprimir
NoMinus:
    cmp al, 41h ;41h = ASCII letra A
    jb Imprimir
    cmp al, 5Ah ;5Ah = ASCII letra Z
    ja Imprimir
    or al, 00100000b ;Convierte 4 ó 5 en 6 ó 7
Imprimir:
    ret
ConvMayMin ENDP
```

OTRA SOLUCIÓN

```
ConvMayMin PROC
    cmp al, 61h ;61h = ASCII letra a
    jb NoMinus
    cmp al, 7Ah ;7Ah = ASCII letra z
    ja NoMinus
    sub al, 20h ;Restamos 20h y pasa a mayúsc.
    jmp Imprimir
NoMinus:
    cmp al, 41h ;41h = ASCII letra A
    jb Imprimir
    cmp al, 5Ah ;5Ah = ASCII letra Z
    ja Imprimir
    add al, 20h ;Sumamos 20h y pasa a minusc.
Imprimir:
    ret
ConvMayMin ENDP
```



Apellidos, Nombre:

Gestión

Sistemas

4) A continuación se muestra el código fuente máquina de un programa en ensamblador tal y como lo muestra CodeVIEW.

4063:000A	0000	ADD	Byte Ptr [BX+SI],AL
4063:000C	0000	ADD	Byte Ptr [BX+SI],AL
4063:000E	0000	ADD	Byte Ptr [BX+SI],AL
INICIO:			
4063:0010	B86540	MOU	AX,4065
4063:0013	8ED8	MOU	DS,AX
4063:0015	33C0	XOR	AX,AX
OTRO:			
4063:0017	FEC0	INC	AL
4063:0019	A20400	MOU	Byte Ptr [0004],AL
4063:001C	3C0A	CMP	AL,0A
4063:001E	75F7	JNZ	OTRO <0017>
4063:0020	B44C	MOU	AH,4C
4063:0022	CD21	INT	21
4063:0024	0A4E42	OR	CL,Byte Ptr [BP+42]
4063:0027	3030	XOR	Byte Ptr [BX+SI],DH
4063:0029	90	NOP	
4063:002A	0000	ADD	Byte Ptr [BX+SI],AL

a) Indíquese cuál es la codificación en lenguaje máquina de la instrucción "JNZ OTRO"

Solución: 75F7h

b) A qué dirección (CS:IP) salta dicha instrucción cuando se cumple la condición que evalúa:

Solución: CS:IP = 4063:0017

c) Sabiendo que el segundo byte de la instrucción indica de algún modo la dirección de salto, indíquese por qué ese segundo byte tiene el valor que ha indicado usted en el apartado a).

Solución: Por que hay que restarle al IP actual 9, para que el desplazamiento resultante sea 0017
 IP actual = 0020, IP del salto = 0017. Hay que sumar -9 (F7h)

d) Si se quisiera saltar a la dirección 4063:0020, ¿cuál sería la nueva codificación de la instrucción en lenguaje máquina?

Solución: 7500h. No hay que restar nada

PUNTUACIÓN DE LA PROGRAMACIÓN EN ENSAMBLADOR

Ejercicio 1: a) 1 punto b) 1 punto

Ejercicio 2: 1 punto

Ejercicio 3: a) 1 punto b) 1 punto c) 1 punto

Ejercicio 4: a) 0,25 puntos b) 0,25 puntos c) 0,25 puntos d) 0,25 puntos