


<div> <div>A</div> <div> Instrucciones generales para la realización de este examen La respuesta debe escribirse en el hueco existente a continuación de cada pregunta con letra clara. Cada respuesta correcta suma un punto Cada respuesta incorrecta, ilegible o vacía no suma ni resta. El total de puntos se dividirá entre el total de preguntas y se multiplicará por 10 para obtener la nota del examen. </div> </div> <div> <p>El siguiente programa para la CPU teórica se encarga de escribir en la pantalla una cadena de texto en una de las 8 posibles filas de las que se compone la pantalla cuando se pulsa en el teclado un número de fila válido. Son válidas únicamente las teclas correspondientes a los números comprendidas entre 0 y 7.</p> <p>El código incluido consta de un programa principal (cuyo comienzo está marcado con la etiqueta comienzo), de una rutina de servicio de interrupción del teclado (rutina_teclado) y de un procedimiento llamado escribe_cadena.</p> <ul style="list-style-type: none"> - El programa principal simplemente instala la rutina de servicio de interrupción de teclado en la tabla de vectores de interrupción. - rutina_teclado lee la tecla pulsada y en el caso de la tecla pulsada sea válida muestra en la pantalla en la fila correspondiente a la tecla pulsada la cadena de texto almacenada en la variable cadena concatenada al código ascii de la tecla pulsada. - El procedimiento escribe_cadena recibe dos parámetros a través de la pila. En el orden en que son apilados antes de la llamada al procedimiento, estos parámetros son: 1) el número de la fila en la que tiene que escribir la cadena en pantalla; 2) la dirección de la cadena a escribir. </div>	<div> A continuación se muestra un ejemplo de salida de pantalla para dicho programa, y el código fuente del programa en ensamblador, al cual le faltan algunas instrucciones: </div> <div>  </div> <div> <pre> ORIGEN 2600h INICIO main .PILA XXX .DATOS cadena VALOR "Escribo en F",0 .CODIGO PROCEDIMIENTO escribe_cadena ; ---1--- PUSH R0 PUSH R1 PUSH R2 PUSH R3 PUSH R4 </pre> </div>	<pre> ; Recuperar en R0 la dirección de la cadena INC R6 INC R6 MOV R0, [R6] ; Recuperar en R1 la fila INC R6 MOV R1, [R6] ; Cargar en R2 el número de columnas ; de la pantalla ; ---2--- ; Cargar en R3 la dirección base de la ; pantalla ; ---3--- ; Registro auxiliar XOR R4, R4, R4 ; Posición de comienzo de la escritura bucle: COMP R1, R4 BRZ calculada ADD R3, R3, R2 DEC R1 JMP bucle calculada: MOV R2, [R0] COMP R2, R4 ; ---4--- MOVB R2, 7 MOV [R3], R2 </pre>
--	---	--

<pre>INC R0 INC R3 JMP calculada final: MOV R1, [R6] MOVL R2, 30h MOVH R2, 0h ADD R1, R1, R2 MOVH R1, 7 MOV [R3], R1 POP R4 POP R3 POP R2 POP R1 POP R0 POP R6 RET FINP PROCEDIMIENTO rutina_teclado PUSH R0 PUSH R1 PUSH R2 PUSH R3 MOVL R0, 27h MOVH R0, 0F9h MOVL R1, 0 ; R1 = mascara MOVH R1, 1 bucle_rut: MOV R2, [R0] AND R2, R2, R1 BRZ no_hay_tecla</pre>	<pre>DEC R0 ; R0 = registro datos MOV R2, [R0] ; R2 = tecla pulsada MOVH R2, 0 ; Eliminar scan INC R0 ; R0 = registro control ; comprobar tecla pulsada MOVH R3, 0 MOVL R3, '0' COMP R2, R3 ; ---5--- MOVH R3 ,0 MOVL R3, '7' COMP R2, R3 BRNC comprobarZF JMP correcto comprobarZF: BRNZ bucle_rut correcto: MOVH R3, 0h MOVL R3, 30h SUB R2, R2, R3 PUSH R2 MOVH R2, BYTEALTO DIRECCION cadena MOVL R2, BYTEBAJO DIRECCION cadena PUSH R2 CALL escribe_cadena INC R7 INC R7 JMP bucle_rut no_hay_tecla: POP R3 POP R2 ; ---6---</pre>	<pre>FINP main: MOVH R0, 0 MOVL R0, 26 MOVH R1, BYTEALTO DIRECCION rutina_teclado MOVL R1, BYTEBAJO DIRECCION rutina_teclado MOV [R0], R1 STI JMP -1 FIN</pre> <p>— Después de haber generado una interrupción y de recibir de la CPU la señal INTA, ¿Cuál es el primer número que la interfaz de teclado envía por el bus de datos? Contestar en hexadecimal</p> <div>001Ah</div> <p>— ¿Qué instrucción o instrucciones falta/n en —1—?</p> <div>PUSH R6 MOV R6, R7</div> <p>— ¿Qué tamaño mínimo debe tener la pila? Contestar en decimal</p> <div>15</div> <p>— ¿Qué instrucción o instrucciones falta/n en —2—?</p>
--	---	--

<div> MOV_L R2, 15 </div> <div> MOV_H R2, 0 </div> <div> — ¿Qué instrucción o instrucciones falta/n en —3—? </div>	<div> BRC bucle_rut </div>	<div> — El dispositivo de memoria ROM se ha construido utilizando chips de 32x4, ¿Cuántas salidas tiene el decodificador de este dispositivo? </div> <div> 16 </div>
<div> MOV_L R3, 80h </div> <div> MOV_H R3, 0F9h </div> <div> — En que direcciones se encuentran mapeados los registros de datos y control del teclado. Contestar en hexadecimal </div> <div> Registro de datos: F926h </div> <div> Registro de control: F927h </div> <div> — ¿Qué instrucción o instrucciones falta/n en —4—? </div>	<div> POP R1 </div> <div> POP R0 </div> <div> IRET </div> <div> ¿En la CPU elemental se han mapeado los siguientes dispositivos: <ul style="list-style-type: none"> Dispositivo RAM: tamaño 16KB, a partir de la dirección C000h. Dispositivo ROM: en el rango AC00 – ADFF. Dispositivo de entrada/salida: tamaño 4KB desde la dirección 7000h. </div> <div> — ¿Qué tamaño del espacio de direcciones del computador que se encuentra libre sin mapear? </div> <div> 43 KB y 512 bytes </div> <div> — El dispositivo de memoria RAM se han construido utilizando chips de 1Kx8, ¿Cuál es el rango de direcciones del banco 1? Nota: los bancos se empiezan a numerar por el cero. </div> <div> C400 – C7FFh </div>	<div> — Dibuja el circuito de activación del dispositivo de entrada/salida. Etiqueta correctamente las líneas. </div> <div> $\sim a_{15} \cdot a_{14} \cdot a_{13} \cdot a_{12}$ </div> <div> — El valor 43004000h representa un número real en formato IEEE-754, ¿a qué número decimal corresponde? </div> <div> 128.25 </div> <div> — Se sabe que el código unicode del carácter versicle (Ÿ) es el U+2123. Codifica este carácter en UTF-8. Responder en hexadecimal. </div> <div> E284A3h </div>
<div> — ¿Cuál es el código ASCII del ‘4’? Contestar en hexadecimal </div> <div> 34h </div> <div> — ¿Qué instrucción o instrucciones falta/n en —5—? </div>		

— Determinar cuáles de las siguientes afirmaciones son CIERTAS. Responder TODAS o NINGUNA si se considera que es la respuesta adecuada.

- A) La instrucción CALL modifica el registro de estado.
- B) El vector de interrupción sirve para determinar a partir de qué dirección se mapea un dispositivo.
- C) La suma de dos números representados en signo-magnitud nunca genera desbordamiento.
- D) El juego de instrucciones de una CPU se puede modificar en función de las necesidades de la aplicación que se quiera programar.

Ninguna

— Determinar cuáles de las siguientes afirmaciones son CIERTAS. Responder TODAS o NINGUNA si se considera que es la respuesta adecuada.

- A) El flag CF sirve para determinar cuando un número entero es mayor que otro.
- B) La secuencia de bits 010 tiene como máximo 4 significados posibles.
- C) La señal de control JUMP se podría utilizar para implementar la instrucción MOVH Rd, Inm8.
- D) Sumar dos números naturales mayores de 2^{15} en una ALU de 16 bits genera carry.

D

— Representar el número 16.51 en un formato de coma fija con 8 bits para la parte entera y 4 para la parte fraccionaria. Responder en hexadecimal.

108h

— ¿Cuál será el menor número representable en este formato si se utiliza complemento a 2 para los números negativos. Responder en decimal.

-128

Sin signo	Relación	Con signo
ZF = 1	=	ZF = 1
ZF = 0	≠	ZF = 0
CF=0 AND ZF=0	>	SF=OF AND ZF=0
CF = 0	≥	SF=OF
CF = 1	<	SF ≠ OF
CF=1 OR ZF=1	≤	SF≠OF OR ZF=1