

Se APRUEBA obteniendo 10 (DIEZ) o más puntos

- 1) Analice cada instrucción e indique si es válida (marcando debajo de ☒) o no lo es (marcando debajo de ☐). Cada respuesta vale 0,5 puntos. Las respuestas correctas suman ese puntaje y las incorrectas lo restan.

☒ ☒ ¿La instrucción es válida?

☐ ☒ SUB BX, OFFSET TABLA (± 0,5p)

☐ ☒ JMP SUBROUTINA (± 0,5p)

☐ ☒ XOR DX, CL (± 0,5p)

☒ ☒ ¿La instrucción es válida?

☒ ☐ SBB CL, CH (± 0,5p)

☐ ☒ MOV [CX], CL (± 0,5p)

☒ ☐ MOV DATO, OFFSET DATO (± 0,5p)

- 2) Escriba la instrucción completa para desapilar el valor almacenado en el tope la pila y dejarlo en el registro AX. (1p)

POP AX

- 3) Si el registro SP contiene el valor 7FE0H, ¿qué valor tendrá tras ejecutar la instrucción CALL LECTURA? (2p)

7FD8

El siguiente programa determina si un año dado en la etiqueta ANIO es o no bisiesto. Al finalizar la ejecución, el programa deja en BISIESTO un 1 si el año era bisiesto o un 0 si no lo era. Recordar que un año es bisiesto si es divisible por 4, excepto aquellos divisibles por 100 pero no por 400. Por ejemplo, el año 2000 fue bisiesto pero el año 1900 no lo fue, a pesar que ambos son divisibles por 4. Adicionalmente, es posible determinar si un número binario es divisible por 4 verificando que sus dos bits menos significativos sean 0. En cambio, para determinar si un número binario es divisible por 100, se puede calcular mediante restas sucesivas el cociente de dividirlo por 100. Si resulta que dicho cociente además posee sus dos bits menos significativos en 0, entonces se puede concluir que el número es divisible por 100 y además por 4, es decir, por 400.

```

1      ORG 1000H
2  ANIO DW 2020
3  ES_BISIESTO DB ?
4
5      ORG 2000H
6      Instruccion a completar
7      AND AX, 0003H
8      JNZ NO_ES_BIS
9      MOV DX, ANIO
10 REPITE: INC AX
11      SUB DX, 100
12      Instruccion a completar
13      JS ES_BIS
14      JMP REPITE
15  TAL_VEZ: AND AX, 0003H
16      JNZ NO_ES_BIS
17  ES_BIS: MOV ES_BISIESTO, 1
18      Instruccion a completar
19  NO_ES_BIS: MOV ES_BISIESTO, 0
20      HLT
21  END

```

- 4) ¿Qué instrucciones faltan en las líneas indicadas para que el programa realice la tarea pedida? (2p)

Línea 6: MOV AX, OFFSET ANIO

Línea 12: JNZ TAL_VEZ

Línea 18: HLT

- 5) ¿Cuántas veces se ejecuta la instrucción SUB DX, 100 (línea 11) al correr el programa? (2p)

22 veces

- 6) ¿A qué dirección de memoria hace referencia la etiqueta ES_BISIESTO? (2p)

ES_BISIESTO = 1005h

- 7) Al finalizar la ejecución del programa dado, ¿qué valor queda guardado en el registro DX? (2p)

DX = -1800

- 8) ¿Cuántas veces se realiza el salto al ejecutar la instrucción JS ES_BIS (línea 13)? (2p)

1 vez

TEMA 00:

Organización de Computadoras – Rec. 3er Parcial

- 1) **XOR DX, CL** Se están mezclando operandos de distinto tipo.
MOV [CX], CL El registro CX no puede ser usado en modo indirecto.
- 2) **POP AX**
- 3) **CALL LECTURA** apila la dirección de retorno (un valor de 16 bits), por lo que:
 $SP \Rightarrow SP - 2 = 7FE0H - 2 = 7FDEH$
- 4) Línea 6: **MOV AX, ANIO**
 Línea 12: **JZ TAL_VEZ**
 Línea 18: **HLT**

```

1      ORG 1000H
2  ANIO    DW 2020
3  ES_BISIESTO DB ?
4
5      ORG 2000H
6      MOV AX, ANIO
7      AND AX, 0003H
8      JNZ NO_ES_BIS
9      MOV DX, ANIO
10 REPITE: INC AX
11      SUB DX, 100
12      JZ TAL_VEZ
13      JS ES_BIS
14      JMP REPITE
15 TAL_VEZ: AND AX, 0003H
16      JNZ NO_ES_BIS
17 ES_BIS: MOV ES_BISIESTO, 1
18      HLT
19 NO_ES_BIS: MOV ES_BISIESTO, 0
20      HLT
21 END

```

- 5) **21 veces:** 1 vez por cada resta de 100 hasta que DX quede negativo.
 $2000 / 100 = 20$ más una vez extra al restar 100 a 20.
- 6) $ES_BISIESTO = 1000H + 2 \text{ bytes de } ANIO = 1002H$
- 7) En DX se calcula el resto de dividir ANIO por 100, restando de a 100 hasta que se llega a 0 o a un valor negativo.
 Como 2020 no es divisible por 100, cuando llega a 20 y resta 100, el resultado será -80, que es Ca2 es:
 $80 = 0000\ 0000\ 0101\ 0000_{B55} \rightarrow -80 = 1111\ 1111\ 1011\ 0000_{Ca2} = FFB0$
- 8) **1 vez:** Se produce cuando se determina que DX no es divisible por 100.