Examen Escrito 2 – Temario A

Nombre: _____ Carné: _____ _____

I. VERDADERO / FALSO (20 puntos – 4 c/u) Marque una X si es Verdadero o Falso. En caso de ser falso, debe explicar por qué, para que su respuesta sea válida.

1. V (x) F( )	La instrucción MOV no permite mover un valor inmediato a un registro de segmento
2. V ( ) F(x)	Bajo ninguna circunstancia, el registro DS puede ser el operando destino de una instrucción MOV Si puede ser, en movimientos registro-registro
3. V ( ) F(x)	La suma de 7FH con 05H en registros de 8 bits, fija la bandera de Carry La suma es 84H, no se sobrepasa
4. V (x) F( )	La instrucción MOV AL, [BX] utiliza direccionamento indirecto
5. V ( ) F(x)	La instrucción MOV [SI], [DI] es correcta Es incorrecta, no se puede hacer movimiento memoria memoria

II. SELECCIÓN MULTIPLE (30 puntos – 10 c/u) Encierre en un círculo la respuesta correcta. Solo UNA.

1. (10 puntos) Seleccione el grupo de instrucciones que implementan la siguiente instrucción en lenguaje ensamblador. No se permite modificar CX o DX:

AX = (DX – CX) – PALABRA + 1

Tome en cuenta que PALABRA está definida así: **PALABRA DW ?**

a.
```
MOV AX, PALABRA
NEG AX          ; instrucción que cambia el signo del operando
SUB DX, CX
ADD AX, DX
INC AX
```

b.
```
MOV AX, PALABRA
NEG AX          ; instrucción que cambia el signo del operando
MOV BX, DX
SUB BX, CX
ADD AX, BX
INC AX
```

c.
```
MOV AX, PALABRA
NEG AX          ; instrucción que cambia el signo del operando
SUB CX, DX
ADD AX, CX
INC AX
```

d.
```
MOV AX, PALABRA
MOV DX, BX
SUB BX, CX
ADD AX, BX
INC AX
```

2. (10 puntos) Implemente la siguiente expresión en lenguaje ensamblador:

ARREGLO[0] = ARREGLO[1] + ARREGLO[2]

Tome en cuenta que ARREGLO está definida así: **ARREGLO DB 10 DUP(?)**

a.
```
MOV AX,[ARREGLO+1]
ADD AX,[ARREGLO+2]
MOV ARREGLO,AX
```

b.
```
MOV AX,[ARREGLO+2]
ADD AX,[ARREGLO+4]
MOV ARREGLO,AX
```

c.
```
MOV AX,[ARREGLO+4]
ADD AX,[ARREGLO+2]
MOV ARREGLO,AX
```

d.
```
MOV AX,[ARREGLO]
ADD AX,[ARREGLO+1]
MOV ARREGLO,AX
```

3. (10 puntos) Indique cuál de los siguientes grupos de instrucciones copia la cadena FUENTE a la cadena DESTINO, si ambas tienen longitud 10 caracteres:

Tome en cuenta que los datos están definidos así:
```
FUENTE    DB    "Cadena Uno"
DESTINO   DB    ?
CAR       DB    ?
```

a.
```
      LEA BX, FUENTE
      LEA DI, DESTINO
      MOV CX, 10
CIC:  MOV DX, [BX]
      MOV [DI], DX
      INC BX
      INC DI
      LOOP CIC
```

b.
```
      LEA BX, FUENTE
      LEA DI, DESTINO
      MOV CX, 10
CIC:  MOV DL, [BX]
      MOV [DI], DL
      INC BX
      INC DI
      LOOP CIC
```

c.
```
      LEA BX, FUENTE
      LEA DI, DESTINO
      MOV CX, 10
CIC:  MOV CAR, [BX]
      MOV [DI], CAR
      INC BX
      INC DI
      LOOP CIC
```

III. COMPLETAR (50 puntos)

1. (10 puntos) Codifique las instrucciones para sumar sucesivamente cada número almacenado en TABLA en el registro AL. Utilice direccionamiento directo, instrucciones ADD y repeticiones.

```
TABLA DB    10,20,30,40,50

            XOR AL, AL
            XOR BX, BX
REPETIR:    ADD AL, TABLA[BX]
            INC BX
            LOOP REPETIR
```

2. (15 puntos) Utilizando operaciones únicamente booleanas (AND, OR, XOR, TEST) y/o desplazamiento de bits, escriba el código en lenguaje ensamblador que ejecute lo que se indica. No se permite utilizar MOV ni operaciones aritméticas.

   a. Colocar el valor cero al registro BX

   ```
   XOR BX, BX
   ```

   b. Dividir entre 8 el valor almacenado en BL

   ```
   MOV CL, 3
   SHR BL, CL      ; cuando es mas de una vez se debe usar CX
   ```

   c. Multiplicar por 2 el valor almacenado en el registro AX

   ```
   SHL AX, 1
   ```

3. (20 puntos) Utilizando operaciones aritméticas, de comparación y de salto, escriba el código en lenguaje ensamblador que ejecute lo que se indica:

   a. (5 puntos) Si la resta entre el registro BX y DX no es cero, ir a SALTO1

   ```
   SUB BX, DX      ; puede ser CMP BX, DX
   JNZ SALTO1      ; puede ser JNE SALTO1
   ```

   b. (5 puntos) Si la multiplicación entre los registros CX y BX genera un acarreo, ir a SALTO2

   ```
   MOV AX, BX
   MUL CX
   JC SALTO2
   ```

   c. (10 puntos) Si el valor almacenado en AX está en el rango de '0' a '9' ir a SALTO3

   ```
            MOV CX, 39H ; compara limite superior
            CMP AX, CX
            JG ESTAMALO
            MOV CX, 31H ; compara limite inferior
            CMP AX, CX
            JL ESTAMALO
   ESTABUENO:     ;bien validado
   ESTAMALO:      ;fuera del rango
   ```

4. (5 puntos) En el siguiente programa que utiliza procedimientos hay un error en tiempo de ejecución. Señálelo claramente e indique cómo lo arreglaría:

```
.MODEL SMALL
.STACK 64
.DATA
;-------------------------------------------------------
.CODE
;----------- Programa principal---------------------------
BEGIN PROC
      MOV AX,@DATA
      MOV DS, AX
      CALL PROC01
      CALL PROC02
      MOV AX, 4CH
      INT 21H
BEGIN ENDP
;-------------------------------------------------------
PROC01 PROC NEAR
      CALL PROC02
      ; ... instrucciones
PROC01 ENDP ; AQUI FALTA EL RET, SINO EL PROCEDIMIENTO NUNCA REGRESA
;-------------------------------------------------------
PROC02 PROC NEAR
      ; ... instrucciones
      JMP PROC01 ; ES INCORRECTO SALTAR A UN PROCEDIMIENTO, MEJOR REGRESAR CON RET
                 ; SINO QUEDARIA UN CICLO INFINITO
PROC02 ENDP
;-------------------------------------------------------
END BEGIN
```