Parcial 2: Módulo A

1) Interprete el código indicado a continuación. Se Pide:

1. Exprese matemáticamente la operación realizada con los operandos.
2. Construya la traza y muestre el resultado final si se ingresan los números 10 y 8 en base decimal.

000 MSG ‘Que hago?’; (?)Incognito.

001 LDT ‘Ingrese un número:’; (?)Incognito.

002 STA 00F; (?)Incognito.

003 LDT ‘Ingrese un número:’; (?)Incognito.

004 NOT Ax; (?)Incognito.

005 ADD 00F; (?)Incognito.

006 JNO 009; (?)Incognito.

007 INC Ax; (?)Incognito.

008 JMP 00D; (?)Incognito.

009 NOT Ax; (?)Incognito.

00A STA 00F; (?)Incognito.

00B CLA; (?)Incognito.

00C SUB 00F; (?)Incognito.

00D EAP ‘...’; ¿Qué mensaje corresponde poner aquí?

00E HLT

00F

2) Escribir el código correcto para ejecutar un programa que cumpla las siguientes condiciones:  
 a) Ingresar una secuencia de números, hasta cargar el número 1.

b) Informar cual de los números ingresados es el mayor.

3) Responda las siguientes preguntas sobre las características de SimuProc:

¿Cuál es la longitud de palabra de cada registro?

¿Qué instrucciones debe utilizar para realizar una comparación de dos números de punto flotante?

¿Cómo funciona la pila provista?

Punto 1 - A y B:

1) Interprete el código indicado a continuación. Se Pide:

1. Exprese matemáticamente la operación realizada con los operandos.
2. Construya la traza y muestre el resultado final si se ingresan los números 10 y 8 en base decimal.

000 MSG ‘Resta por complemento?’;

001 LDT ‘Ingrese un número:’;

002 STA 00F;

003 LDT ‘Ingrese un número:’;

004 NOT Ax;

005 ADD 00F;

006 JNO 009;

007 INC Ax;

008 JMP 00D;

009 NOT Ax;

00A STA 00F;

00B CLA;

00C SUB 00F;

00D EAP ‘El resultado de la resta es:’;

00E HLT

00F

**Realizamos el punto 1 y 2 en este ejercicio:**

| **orden** | **Instrucción** | **Ax** | **00F** | **Bx** | **OverFlow Flag** | **Que hace?** |
| --- | --- | --- | --- | --- | --- | --- |
| **000** | **MSG** |  |  |  | **0** | **Imprimir “¿Que hago?”** |
| **001** | **LDT** | **10** |  |  | **0** | **Pide a usuario número: (10) // Ax:= 10** |
| **002** | **STA 00F** | **10** | **10** |  | **0** | **00F:=Ax // Guardo Ax en 00F** |
| **003** | **LDT** | **8** | **10** |  | **0** | **Pide a usuario número: (8) // Ax:= 8** |
| **004** | **NOT Ax** | **8’** | **10** |  | **0** | **Alterno 8 a 8’ = (1111111111110111)** |
| **005** | **ADD 00F** | **1** | **10** | **1** | **1** | **8’+10= 10000000000000001= 1** |
| **006** | **JNO 009** | **1** | **10** | **1** | **1** | **Como SI está activado el flag de OverFlow, no salta.** |
| **007** | **INC Ax** | **2** | **10** | **1** | **1** | **Incrementamos a +1 Ax// Ax:= 1+1= Ax:=2** |
| **008** | **JMP 00D** | **2** | **10** | **1** | **1** | **Salta a 00D** |
| **00D** | **EAP** | **2** | **10** | **1** | **1** | **Imprime el resultado de Ax(2)** |
| **00E** | **HLT** | **2** | **10** | **1** | **0** | **Finaliza el Programa.** |

Punto 2 - A y B:

2) Escribir el código correcto para ejecutar un programa que cumpla las siguientes condiciones:  
 a) Ingresar una secuencia de números, hasta cargar el número 1.

b) Informar cual de los números ingresados es el mayor.

**#SimuProc 1.4.2.0**

**#Ejercicio del Parcial 2:**

**CLA**

**MSG "Bienvenido una vez más Amo mío";**

**MSG "Por favor, ingrese números";**

**MSG "Pero no ingrese el 1, que si lo hace mi proceso terminará";**

**MSG "Si lo hace, igualmente, le dire cual es el número más grande que me dio";**

**MSG "Juguemos una vez más amo mío";**

**LDT "Pero recuerde, el 1 me corta el proceso:";**

**CMP 021;Primero comparo con la Ancla, es igual a 1? Para salir.**

**JEQ 00E;Si es 1, salta a 00D**

**CMP 020;Ahora comparo con la var más grande.**

**JMA 01C;Si es más grande se va a 019.**

**LDT "Ingrese otro numero Amo mío:"; Sino pido otro número.**

**JMP 007; y repito el proceso infinito.**

**#00E**

**MSG "Amo... Porque...";**

**MSG "porque NO QUIERES JUGAR MÁS CONMIGO!!!!";**

**MSG "...";**

**MSG "El número más grande que me dio usted Amo mío";**

**LDA 020; Ax:= 020 cargo en Ax la variable más grande.**

**EAP "Fue el número: ";**

**MSG "Espero que se haya divertido tanto como yo...";**

**MSG "Hasta luego... Amo mio.";**

**MSG "Lo voy a extrañar...";**

**MSG "Proceso terminado.";**

**MOV 020,021; limpio la variable.**

**CLA; Limpio Ax.**

**HLT;**

**#01C**

**STA 020; 020:=Ax Ahora es el número más grande.**

**LDT "Ingrese otro numero Amo mio:"; pido otro número**

**JMP 007;Vuelvo a comparar.**

**#020**

**1; La Variable "Más Grande"**

**#021**

**1; La Variable "Ancla"**

Punto 3 - A, B y C:

3) Responda las siguientes preguntas sobre las características de SimuProc:

1. ¿Cuál es la longitud de palabra de cada registro?
2. ¿Qué instrucciones debe utilizar para realizar una comparación de dos números de punto flotante?
3. ¿Cómo funciona la pila provista?

**a)   
  
Documentación:** La Memoria es el dispositivo que almacena toda la información del programa que se ejecuta, tanto datos como instrucciones.

Ésta en realidad no es parte del procesador, sino que es un dispositivo aparte al que el

procesador accede para ir leyendo las instrucciones y datos del programa.   
La capacidad de la memoria Simulada es de 4096 posiciones de 16 bits cada registro:

Desde 000 hasta FFF.

**Short Answer:** La longitud de palabra de cada registro es de 16 bits.

**b)** Después de buscar en la documentación, y no encontrar una instrucción exclusiva del punto flotante, creería que la respuesta es CMP.

**c)**

**Documentación:**

Registros de Pila:

**BP:** Base Pointer, Puntero de base de la pila. El valor de por defecto es F80, Este puede

cambiarse desde un programa, asignándole otra dirección de memoria con la instrucción MOV. Digamos que quiero reservar más espacio para la pila haciendo que ésta comience desde la posición CF1, entonces copio esta dirección de memoria en cualquier posición de memoria; supongamos que lo copie en la dirección 3B entonces uso la instrucción MOV BP, 3B y así BP es igual a CF1. Mientras se ejecuta el programa se puede visualizar en una barra de porcentaje el uso de la pila.

**SP:** Stack Pointer, Puntero de la pila, indica en qué próxima dirección de la pila está disponible, es decir, apunta a la cima de la pila. Este valor se cambia automáticamente cuando se usan las instrucciones PUSH POP.

**Short Answer:** Básicamente la pila provista cuenta de 2 partes, una que indica que tan larga es, que por defecto es de F80 espacios de memoria, y la otra indica en qué próxima dirección de la pila está disponible, es decir, apunta a la cima de la pila. Este valor se cambia automáticamente cuando se usan las instrucciones PUSH POP.