

### Ejercicio #1.

Desarrollar un programa en lenguaje ensamblador para arquitectura Intel x86 que calcule la siguiente expresión:

$$w(x, y, z) = x + y - z + 4$$

Donde  $x, y, z, w$  son variables de tipo byte; y están inicializadas con  $x = 35h, y = 23h, z = 18h$

#### Qué debe hacer el programa:

Definir en segmento de datos las variables  $w, x, y, z$  de tipo byte. A las variables  $x, y, z$  asignar el valor correspondiente. La variable  $w$  puede inicializarse en 0 o indefinido (?) ya que al final se modificará para guardar el valor de la ecuación.

El programa deberá ser capaz de calcular la expresión utilizando las instrucciones ADD y SUB. Una vez calculada dicha expresión, el resultado debe almacenarse en la variable  $w$  en el segmento de datos.

#### Consideraciones:

- Tomar en cuenta que las variables son de tamaño byte y el resultado cabe perfectamente en un dato de tipo byte.
- Ignorar el registro de banderas.
- En el Debugger se deberá visualizar el contenido del segmento de datos con las variables definidas, al igual que el resultado una vez que se calcula y se almacena.

#### Entregables:

Me deberán enviar por correo el código fuente (archivo .asm). La fecha límite de entrega es el día 18 de marzo de 2020, antes de las 12:00 hrs. Archivos recibidos después de ese horario no se tomarán en cuenta.

## Ejercicio #2.

Desarrollar un programa en lenguaje ensamblador para arquitectura Intel x86 que calcule la suma de dos números hexadecimales definidos a continuación:

*num1*: A637 548B E1C0h

*num2*: DF76 F250 C193h

Ambos números deberán estar definidos en el segmento de datos. Cada número hexadecimal está dividido en 3 datos de tipo word. El resultado, si bien se observa, deberá almacenarse en 4 words; o bien, 3 words y una localidad extra para almacenar el acarreo generado, dentro del segmento de datos.

### Qué debe hacer el programa:

Definir en segmento de datos las variables *num1*, *num2* y *res*, de tipo word. Cada variable estará dada por 3 datos de tipo word (en el caso de *res*, podría definirse con 4 datos).

El programa deberá ser capaz de calcular la suma de *num1* y *num2* haciendo uso de la instrucción ADC y almacenar el resultado en la variable *res*. Cada número (*num1* y *num2*) deberá separarse en bloques de tamaño de 1 word. Sumar el primer bloque (el menos significativo, el de la extrema derecha) de cada número y almacenar el resultado en el primer bloque de *res*. Para el segundo bloque (el que sigue a la izquierda del primero), hacer la suma considerando el acarreo de la suma anterior, y almacenarlo en el segundo bloque de *res*. Para el tercer bloque ocurrirá algo similar. Por último, si se genera acarreo de la suma de los 3 bloques, almacenarlo en un cuarto bloque de la variable *res*.

### Consideraciones:

- En el segmento de datos, la parte menos significativa de cada número se almacena primero y la parte más significativa al final.
- Si se accede [*num1* + 1], recordar que *num1* apunta al primer byte de la variable, y por lo tanto, *num1* + 1 hace referencia al segundo byte.
- En el Debugger se deberá visualizar el contenido del segmento de datos con las variables definidas, al igual que el resultado una vez que se calcula y se almacena.

### Entregables:

Me deberán enviar por correo el código fuente (archivo .asm). La fecha límite de entrega es el día 18 de marzo de 2020, antes de las 13:00 hrs. Archivos recibidos después de ese horario no se tomarán en cuenta.