**TP - Módulo de Programación en bajo nivel - 4 de noviembre de 2019**

**Forma de entrega: Guardar el código modificado en un documento NombreApellidoAssembler.txt (dónde ApellidoNombre es el apellido y nombre del alumno) y enviar por mensajería al ayudante del grupo y al JTP.**

1. Complete la subrutina **POT** en los espacios señalados con \_\_ y con las instrucciones que considere necesarias para que sea reusable. La subrutina debe recibir un parámetro **N** por referencia a través de la pila y calcular la N-ésima potencia de otro parámetro **X** pasado por valor a través de registro (XN). El resultado debe verse reflejado en el parámetro recibido por referencia. Los datos son de tipo Word y enteros positivos. **Nota:** no modifique la subrutina MULTI1 (reemplaza la operación MUL no disponible en VonSim).

**ORG \_\_\_**

**POT:** MOV BX, SP

ADD BX, \_\_

; Recibir parámetros

MOV DX, 1

ITER: CMP CX, 0

JZ FINR

**CALL** **MULT1**

DEC CX

JMP ITER

FINR: MOV \_\_, DX ; Guardo

**RET**

**ORG 3500H**

**MULT1:** PUSH BX

PUSH CX

MOV BX, AX

MOV CX, 0

SIGUE: CMP BX, 0

JZ FFIN

ADD CX, DX

DEC BX

JMP SIGUE

FFIN: MOV DX, CX

POP CX

POP BX

**RET**

1. Complete el siguiente programa en Assembler en los espacios señalados con \_\_ y con las instrucciones que considere necesarias para que, **utilizando la subrutina POT**, calcule las potencias detalladas en la TABLA para el valor almacenado en la variable X, y los almacene en la misma tabla. Además, el programa debe sumar los resultados de las potencias y almacenar dicha suma en memoria.

**ORG \_\_**

TABLA \_\_ 1, 2, 5, 4, 3

X \_\_ 3

SUM \_\_ 0

**ORG \_\_**

MOV BX, OFFSET NPOT

MOV AX, X

MOV CX, \_\_

MOV DX, \_\_

BUCLE: CMP CX, 0

J\_\_ FIN

; Enviar parámetros

**CALL POT**

; Volver de la subrutina

ADD DX, [BX] ; Suma

ADD BX, \_\_

SUB CX, \_\_

JMP BUCLE

FIN: MOV SUM, \_\_

HLT

END