

## Programa 2: Ensamblador de un solo paso

Código fuente: *matrícula\_p2.c / matrícula\_p2.cpp / matrícula\_p2.java*

Este programa consiste en el ensamblado en un sólo paso del código ensamblador contenido en un archivo de texto. La especificación del lenguaje ensamblador a tomar en cuenta es la siguiente:

Características generales de la arquitectura	
Propiedad	Valor
Unidad mínima direccionable (byte)	8 bits
Tamaño de una palabra	16 bits
Tamaño de una dirección	16 bits
Codificación de datos	Little endian a nivel de bytes
Codificación de enteros	Complemento a dos

Registros disponibles	
Nombre del registro	Código máquina
R0	000
R1	001
R2	010
R3	011
R4	100
R5	101
RC	110
RP	111

Conjunto de instrucciones	
Sintaxis	Código máquina
NOP	00000000
ADD <i>ra, rb, rc</i>	0001000 <i>aaabbbccc</i>
SUB <i>ra, rb, rc</i>	0001001 <i>aaabbbccc</i>
MUL <i>ra, rb, rc</i>	0001010 <i>aaabbbccc</i>
DIV <i>ra, rb, rc</i>	0001011 <i>aaabbbccc</i>
REM <i>ra, rb, rc</i>	0001100 <i>aaabbbccc</i>
AND <i>ra, rb, rc</i>	0001101 <i>aaabbbccc</i>
IOR <i>ra, rb, rc</i>	0001110 <i>aaabbbccc</i>
XOR <i>ra, rb, rc</i>	0001111 <i>aaabbbccc</i>
NEG <i>ra, rb</i>	0000010000 <i>aaabbb</i>

COM <i>ra, rb</i>	0000010001 <i>aaabbb</i>
LST <i>ra, rb</i>	0000010010 <i>aaabbb</i>
GRT <i>ra, rb</i>	0000010011 <i>aaabbb</i>
LSE <i>ra, rb</i>	0000010100 <i>aaabbb</i>
GRE <i>ra, rb</i>	0000010101 <i>aaabbb</i>
EQU <i>ra, rb</i>	0000010110 <i>aaabbb</i>
NEQ <i>ra, rb</i>	0000010111 <i>aaabbb</i>
MOV <i>ra, rb</i>	0000011000 <i>aaabbb</i>
MOV <i>ra, entero</i>	00001 <i>aaae</i> eeeeeeeeeeeeeeee
MOV <i>ra, WORD [entero]</i>	00100 <i>aaae</i> eeeeeeeeeeeeeeee
MOV <i>ra, BYTE [entero]</i>	00101 <i>aaae</i> eeeeeeeeeeeeeeee
MOV WORD [ <i>entero</i> ], <i>ra</i>	00110 <i>aaae</i> eeeeeeeeeeeeeeee
MOV BYTE [ <i>entero</i> ], <i>ra</i>	00111 <i>aaae</i> eeeeeeeeeeeeeeee
MOV <i>ra, WORD [rb]</i>	0100000000 <i>aaabbb</i>
MOV <i>ra, BYTE [rb]</i>	0100000001 <i>aaabbb</i>
MOV WORD [ <i>ra</i> ], <i>rb</i>	0100000010 <i>aaabbb</i>
MOV BYTE [ <i>ra</i> ], <i>rb</i>	0100000011 <i>aaabbb</i>
MOV <i>ra, WORD [rb + entero]</i>	0100000100 <i>aaabbb</i> eeeeeeeeeeeeeeee
MOV <i>ra, BYTE [rb + entero]</i>	0100000101 <i>aaabbb</i> eeeeeeeeeeeeeeee
MOV WORD [ <i>ra + entero</i> ], <i>rb</i>	0100000110 <i>aaabbb</i> eeeeeeeeeeeeeeee
MOV BYTE [ <i>ra + entero</i> ], <i>rb</i>	0100000111 <i>aaabbb</i> eeeeeeeeeeeeeeee
JMP <i>entero</i>	10000010eeeeeeeeeeeeeeee
JIF <i>entero</i>	10000011eeeeeeeeeeeeeeee
JMP <i>ra</i>	11000 <i>aaa</i>
JIF <i>ra</i>	11001 <i>aaa</i>

<b>Ejemplo de sintaxis y uso de etiquetas</b>
eti1: JMP eti1

El archivo de entrada contendrá varias líneas de texto con una instrucción válida por línea. La dirección de la primera instrucción dada es la dirección 0. En caso de que la instrucción contenga algún entero, éste podrá ser codificado siempre en 16 bits. Los enteros negativos son válidos.

Las etiquetas podrán ser usadas en cualquier instrucción que requiera un entero y deberán estar declaradas antes de su uso. Éstas contendrán únicamente caracteres alfanuméricos y comenzarán con un carácter alfabético. El lenguaje ensamblador distingue minúsculas y mayúsculas. La línea de texto no contendrá espacios en blanco excepto en los siguientes casos:

- El espacio en blanco entre el inicio de la instrucción actual y los dos puntos de la etiqueta asociada, si es que existe.
- El espacio en blanco que separa el nombre de la instrucción de la lista de operandos, si es que esta última existe.
- El espacio en blanco que separa la palabra WORD o BYTE del corchete que abre inmediatamente después.

**Entrada:** Una secuencia de a lo mucho 100 líneas de texto. Puede suponer que cada línea tendrá una longitud máxima de 100 caracteres. La entrada deberá ser leída desde el archivo `entrada.txt`

**Salida:** Una secuencia de caracteres 0 ó 1 que denoten el ensamblado del archivo de entrada. La secuencia podrá contener opcionalmente saltos de línea. La salida deberá ser escrita al archivo `salida.txt`

Ejemplo de entrada ( <code>entrada.txt</code> )	Ejemplo de salida ( <code>salida.txt</code> )
NOP ADD R0,R1,R2	000000000001000000001010
NOP eti: JMP eti	00000000100000100000000100000000

Su programa sólo debe imprimir lo solicitado. El código fuente deberá ser enviado como archivo adjunto al correo [racc@correo.azc.uam.mx](mailto:racc@correo.azc.uam.mx) con copia para [rccuam2013@gmail.com](mailto:rccuam2013@gmail.com). No se recibirán ejecutables y de ninguna otra forma.