

Programa 1: Ensamblado de instrucción

Código fuente: *matrícula_p1.c / matrícula_p1.cpp / matrícula_p1.java*

Este programa consiste en el ensamblado de la instrucción dada en una línea 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>	0000010010aaabbb
GRT <i>ra, rb</i>	0000010011aaabbb
LSE <i>ra, rb</i>	0000010100aaabbb
GRE <i>ra, rb</i>	0000010101aaabbb
EQU <i>ra, rb</i>	0000010110aaabbb
NEQ <i>ra, rb</i>	0000010111aaabbb
MOV <i>ra, rb</i>	0000011000aaabbb
MOV <i>ra, entero</i>	00001aaaeeeeeeeeeeeeeeee
MOV <i>ra, WORD [entero]</i>	00100aaaeeeeeeeeeeeeeeee
MOV <i>ra, BYTE [entero]</i>	00101aaaeeeeeeeeeeeeeeee
MOV WORD [<i>entero</i>], <i>ra</i>	00110aaaeeeeeeeeeeeeeeee
MOV BYTE [<i>entero</i>], <i>ra</i>	00111aaaeeeeeeeeeeeeeeee
MOV <i>ra, WORD [rb]</i>	0100000000aaabbb
MOV <i>ra, BYTE [rb]</i>	0100000001aaabbb
MOV WORD [<i>ra</i>], <i>rb</i>	0100000010aaabbb
MOV BYTE [<i>ra</i>], <i>rb</i>	0100000011aaabbb
MOV <i>ra, WORD [rb + entero]</i>	0100000100aaabbbeeeeeeeeeeeeeeee
MOV <i>ra, BYTE [rb + entero]</i>	0100000101aaabbbeeeeeeeeeeeeeeee
MOV WORD [<i>ra + entero</i>], <i>rb</i>	0100000110aaabbbeeeeeeeeeeeeeeee
MOV BYTE [<i>ra + entero</i>], <i>rb</i>	0100000111aaabbbeeeeeeeeeeeeeeee
JMP <i>entero</i>	10000010eeeeeeeeeeeeeeee
JIF <i>entero</i>	10000011eeeeeeeeeeeeeeee
JMP <i>ra</i>	11000aaa
JIF <i>ra</i>	11001aaa

La línea de texto de entrada contendrá una instrucción válida. En caso de que la instrucción contenga algún entero, éste podrá ser codificado siempre en 16 bits. Los enteros negativos son válidos.

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 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 línea de texto. Puede suponer que la línea tendrá una longitud máxima de 100 caracteres.

Salida: Una secuencia de caracteres 0 ó 1 que denoten el ensamblado de la instrucción dada. La secuencia podrá contener opcionalmente saltos de línea.

Ejemplo de entrada	Ejemplo de salida
NOP	00000000
ADD R0,R1,R2	0001000000001010
MOV R0,WORD [1]	001000000000000100000000

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