

Apellido y Nombre Legajo

1.a. Traduzca la siguiente pseudo instrucción a un código equivalente en lenguaje ensamblador, pero que no utilice pseudo instrucciones:

li \$t0, 0xABCOF789	Res:
---------------------	------

1.b. Exprese el siguiente fragmento en lenguaje ensamblador MIPS. Para la construcción if then else utilice únicamente las instrucciones slt, beq, bne. largo y ancho son variable globales.

1.c. Resuelva con algunas instrucciones de MIPS el siguiente requerimiento (puede utilizar pseudo instrucciones):

En la dirección de memoria principal 0x200 se encuentra una media palabra. - Si el bit 5 de esa media palabra tiene valor 1 coloque el valor 0xFFFFFFFF en la dirección 0x204. - Si el bit 5 de esa media palabra tiene valor 0 coloque el valor 0x44444444 en la dirección 0x204	Res:
---	------

1. d. Decodifique las siguientes instrucciones máquina. Indique tambien de qué formato es la instrucción. (ver 2da pagina)

	Instrucción en ensamblador	Formato
0x8dae0004		
0x01a6c024		
0xa1b80000		

2. Una CPU MIPS cuenta con una caché para las INSTRUCCIONES MÁQUINA que ejecuta. Es decir, cada vez que la CPU debe obtener una instrucción y ejecutarla, verifica primero si está en esta caché de instrucciones.

La misma es de mapeo directo de 8 entradas, y en cada entrada se almacenan 4 palabras de datos. La CPU MIPS funciona con un clock de 1Khz (mil ciclos de reloj por segundo).

El acceso a caché toma 1 ciclo de reloj. El acceso a memoria principal toma 50 ciclos de reloj.

El acceso a caché toma 1 ciclo de reloj. El acceso a memoria principal toma 5 ciclos de reloj. Las siguientes instrucciones se encuentran en código máquina en la dirección 0x4000. Indique

<pre> addi \$t0, \$zero, 5000 addi \$t1, \$zero, \$zero loop: beq \$t0, \$zero, salir addi \$t1, \$t1, 20 addi \$t0, \$t0, -1 j loop salir: sw \$t1, 1000(\$zero) </pre>	<p>Cant. de accesos a memoria:</p> <p>Cant. de aciertos:</p> <p>Cant. de fallos:</p> <p>Tasa de aciertos: Tasa de Fallos:</p> <p>Tiempo medio de acceso : (con su ecuación)</p>
--	---

Apellido y Nombre Legajo

3. Traduzca, como lo haría un compilador, la siguiente función, a lenguaje ensamblador MIPS.

```
int sumatoria (int n)
{
    unsigned char temp[100];           // cada elemento de temp ocupa un byte
    int res;   int i;

    for (i=0; i<100; i++)
        temp[i] = random(i);

    for (i=0; i<n; i++) {
        if (temp[i] != 0)
            res = res + temp[i];
    }
    return res;
}
```

Extracto del MANUAL de MIPS:

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">31</td><td style="width: 25%;">26 25</td><td style="width: 20%;">21 20</td><td style="width: 15%;">16 15</td><td style="width: 10%;">0</td></tr> <tr> <td>SB</td><td>base</td><td>rt</td><td>offset</td><td></td></tr> <tr> <td>101000</td><td>6</td><td>5</td><td>5</td><td>16</td></tr> </table>	31	26 25	21 20	16 15	0	SB	base	rt	offset		101000	6	5	5	16	Format: SB rt, offset(base) MIPS32 (MIPS I)									
31	26 25	21 20	16 15	0																					
SB	base	rt	offset																						
101000	6	5	5	16																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">31</td><td style="width: 25%;">26 25</td><td style="width: 20%;">21 20</td><td style="width: 15%;">16 15</td><td style="width: 10%;">0</td></tr> <tr> <td>SLTI</td><td>rs</td><td>rt</td><td>immediate</td><td></td></tr> <tr> <td>001010</td><td>6</td><td>5</td><td>5</td><td>16</td></tr> </table>	31	26 25	21 20	16 15	0	SLTI	rs	rt	immediate		001010	6	5	5	16	Format: SLTI rt, rs, immediate MIPS32 (MIPS I)									
31	26 25	21 20	16 15	0																					
SLTI	rs	rt	immediate																						
001010	6	5	5	16																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">31</td><td style="width: 25%;">26 25</td><td style="width: 20%;">21 20</td><td style="width: 15%;">16 15</td><td style="width: 10%;">0</td></tr> <tr> <td>LW</td><td>base</td><td>rt</td><td>offset</td><td></td></tr> <tr> <td>100011</td><td>6</td><td>5</td><td>5</td><td>16</td></tr> </table>	31	26 25	21 20	16 15	0	LW	base	rt	offset		100011	6	5	5	16	Format: LW rt, offset(base) MIPS32 (MIPS I)									
31	26 25	21 20	16 15	0																					
LW	base	rt	offset																						
100011	6	5	5	16																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">31</td><td style="width: 25%;">26 25</td><td style="width: 20%;">21 20</td><td style="width: 15%;">16 15</td><td style="width: 10%;">0</td></tr> <tr> <td>ADDI</td><td>rs</td><td>rt</td><td>immediate</td><td></td></tr> <tr> <td>001000</td><td>6</td><td>5</td><td>5</td><td>16</td></tr> </table>	31	26 25	21 20	16 15	0	ADDI	rs	rt	immediate		001000	6	5	5	16	Format: ADDI rt, rs, immediate MIPS32 (MIPS I)									
31	26 25	21 20	16 15	0																					
ADDI	rs	rt	immediate																						
001000	6	5	5	16																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">31</td><td style="width: 25%;">26 25</td><td style="width: 20%;">21 20</td><td style="width: 15%;">16 15</td><td style="width: 10%;">0</td></tr> <tr> <td>ORI</td><td>rs</td><td>rt</td><td>immediate</td><td></td></tr> <tr> <td>001101</td><td>6</td><td>5</td><td>5</td><td>16</td></tr> </table>	31	26 25	21 20	16 15	0	ORI	rs	rt	immediate		001101	6	5	5	16	Format: ORI rt, rs, immediate MIPS32 (MIPS I)									
31	26 25	21 20	16 15	0																					
ORI	rs	rt	immediate																						
001101	6	5	5	16																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">31</td><td style="width: 25%;">26 25</td><td style="width: 20%;">21 20</td><td style="width: 15%;">16 15</td><td style="width: 10%;">11 10</td><td style="width: 5%;">6</td><td style="width: 5%;">5</td><td style="width: 10%;">0</td></tr> <tr> <td>SPECIAL</td><td>rs</td><td>rt</td><td>rd</td><td>0</td><td>00000</td><td>AND</td><td>100100</td></tr> <tr> <td>000000</td><td>6</td><td>5</td><td>5</td><td>5</td><td>5</td><td>6</td><td></td></tr> </table>	31	26 25	21 20	16 15	11 10	6	5	0	SPECIAL	rs	rt	rd	0	00000	AND	100100	000000	6	5	5	5	5	6		Format: AND rd, rs, rt MIPS32 (MIPS I)
31	26 25	21 20	16 15	11 10	6	5	0																		
SPECIAL	rs	rt	rd	0	00000	AND	100100																		
000000	6	5	5	5	5	6																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">31</td><td style="width: 25%;">26 25</td><td style="width: 20;"></td><td style="width: 15;"></td><td style="width: 10%;">0</td></tr> <tr> <td>JAL</td><td></td><td>instr_index</td><td></td><td></td></tr> <tr> <td>000011</td><td>6</td><td>26</td><td></td><td></td></tr> </table>	31	26 25			0	JAL		instr_index			000011	6	26			Format: JAL target MIPS32 (MIPS I)									
31	26 25			0																					
JAL		instr_index																							
000011	6	26																							