

# ESTRUCTURA DE COMPUTADORES I

## TEMA 2: EL REPERTORIO DE INSTRUCCIONES

1. ¿Cuál es el código ensamblador MIPS para cargar la siguiente constante de 32 bits en el registro \$18?

0000 0000 0011 1101 0000 1001 0000 0000

2. En el registro \$10 se encuentra almacenado el valor 0x1001000c. ¿Es correcta la instrucción **lw \$8, 3(\$10)**? ¿Por qué?
3. Si en la posición de memoria  $A$  se encuentra almacenado la palabra  $A + 2$  ¿Qué valor se almacena en el registro \$8 después de ejecutar la instrucción **lb \$8, 1(\$10)**, sabiendo que en el registro \$10 se encuentra almacenado el valor 0xff001000?
4. ¿Qué valor tienen los registros \$8 y \$10 después de ejecutar la instrucción **sw \$8, 16(\$10)**, sabiendo que inicialmente \$8 y \$10 tienen almacenado los valores 0xff001000 y 0x10010004, respectivamente? ¿Qué posición de memoria se modifica?
5. La dirección de la instrucción **beq \$18, \$19, Etiqueta** es 0x10a0c300. Contesta a las siguientes cuestiones:
  - a) ¿cuál es la dirección de memoria mayor a la que puede saltar? Si queremos saltar a una dirección mayor ¿por qué secuencia de instrucciones tenemos que reemplazar la instrucción anterior?
  - b) ¿A qué posición en concreto se bifurcará en el caso que el campo desplazamiento de la instrucción sea 0xf01f?
6. La instrucción **j Etiqueta** está en la posición de memoria 0x10010014.
  - a) ¿cuál es la dirección de memoria mayor a la que puede saltar?
  - b) ¿Cuál es el formato de la instrucción e indica el valor de cada campo si el valor de **Etiqueta** es 0x10ffcc84?
7. ¿Qué *string* está almacenado en la dirección de memoria 0x10010000 que contiene el valor 0x45433100?
8. Traducir el siguiente procedimiento en C a lenguaje ensamblador MIPS:

```
void ImageBack(int ancho, int alto, int **raster)
{
    nx = ancho;
    ny = alto;

    for(int i=0; i< nx; i++)
        for(int j=0; j< ny; j++)
            raster[i][j] = 0;
}
```

9. Traducir el siguiente procedimiento en C a lenguaje ensamblador MIPS:

```
int ImageBack(int long, int *vector)
{
    j = 0;
    nx = long;

    for(int i=0; i< nx; i++)
        j = j + vector[i];

    return j;
}
```