



Dadas las siguientes declaraciones de la sección de datos, y sabiendo que el segmento de datos se carga a partir de la dirección de memoria `0x0100 0000`, y el segmento de texto a partir de la dirección `0x0400 0000`.

```
.data
lastLogin:
    .word 1631216321
datos:
    .byte 37
    .half 0x7FFF
    .float 0
nick:
    .asciiz "peperina"
#password (esto es un comentario, no una etiqueta)
    .word 0x68756E74, 0x65723200, 0x00000000
saludo:
    .asciiz "Buen dia <nick>!"
longitud:
    .word 0
```

1. ¿Cuántos bytes son reservados por el segmento de datos? ¿Cuántos son desperdiciados por alineamiento?
2. ¿Cuál es el desplazamiento del primer byte de la contraseña con respecto a la etiqueta ``datos``? ¿Cuál es la dirección del carácter `!"` en ``saludo``?
3. ¿Cuáles serán los contenidos de los registros `t1`, `t2`, y `t3` luego de ejecutar las siguientes instrucciones? ¿Cuáles son las direcciones efectivas de los datos cargados para cada instrucción de carga?

```
li $t0, 4
lb t1, lastLogin+3($t0)
lbu t3, lastLogin+3($t0)
addi $t0, $t0, 4
lw t2, lastLogin($t0)
la $t0, datos
```

4. ¿Cuáles de las instrucciones ensamblador del programa del punto 3 son pseudo instrucciones y cuales son instrucciones reales?
5. Sea una máquina que ejecuta una instrucción cada milisegundo, y cada pseudo instrucción se traduce exactamente a dos instrucciones máquina ¿Cuánto tiempo tardará en ejecutarse la totalidad del código del punto 3?

Ejercicios extra:

1. Traduzca la siguiente estructura de un lenguaje de alto nivel a directivas de ensamblador de mips. Respete el orden de las variables. Tenga en cuenta que el string contiene un carácter con valor cero para indicar el final de la cadena. El tipo short es un entero de 16 bits.

```
struct {  
    float zoom = 1.70;  
    string nombre = "de_dust";  
    short alto = 1024;  
    short ancho = 5012;  
    byte pol = 6;  
    byte puntaje = 0;  
    int key1 = 0x63346133  
    int key2 = 0x00AABBCC  
} mapa;
```

2. Represente un vuelco de memoria en hexadecimal (es decir, los contenidos de la memoria en hexadecimal) del segmento de datos que contiene a la estructura. Complete el espacio que se desperdicia por el alineamiento de los datos con ceros.
3. Reordenar la estructura para que se desperdicie la menor cantidad de espacio por alineamiento ¿Cuántos bytes de diferencia hay entre ambas versiones de la estructura?
4. Utilizando las instrucciones `sw` y `lw` escriba un pequeño programa que copie el valor de las variables `key1` y `key2` a `nombre`.
5. ¿Cuál será el valor de los registros `t0`, `t1` y `t2` luego de ejecutar el siguiente conjunto de instrucciones?

```
lb $t0, pol  
addi $t2, $t0, 2  
lbu $r2, ancho+1($t2)  
li $t2, 4  
lw $t2, zoom($t2)
```

6. ¿Cuáles de las instrucciones ensamblador del programa del punto 5 son pseudo instrucciones y cuales son instrucciones reales?
7. Sea una máquina que ejecuta una instrucción cada milisegundo, y cada pseudo instrucción se traduce exactamente a dos instrucciones máquina ¿Cuánto tiempo tardará en ejecutarse la totalidad del código del punto 5?