



HOJA DE PROBLEMAS DEL TEMA 4. PROGRAMACIÓN EN ENSAMBLADOR DE MIPS NÚMEROS ENTEROS, DATOS BOOLEANOS Y MÁSCARAS DE BITS

1. Realizar un programa en ensamblador que evalúe la siguiente expresión:

$$w = x - y * (z + 2)$$

A continuación se muestra una posible solución en C:

```
int x, y, z, w;
int main (void) {
    w = x - y * (z+2);
    return 0;
}
```

Las variables **x**, **y**, **z** y **w** deben existir en la solución con el mismo nombre, y tendrán un valor inicial desconocido. Para realizar las pruebas se introducirán manualmente los valores necesarios en las variables. Escribir la solución en un fichero llamado **expresion_enteros.asm**.

2. Sea una función **f(x)** y sean dos puntos de la misma con coordenadas **(x₁, y₁)** y **(x₂, y₂)**. El valor de la función en un cierto punto **x_p** tal que **x₁ < x_p < x₂** se puede interpolar mediante la fórmula siguiente:

$$f(x_p) = \frac{(x_p - x_1) \cdot (y_2 - y_1)}{x_2 - x_1} + y_1$$

Se pide realizar un programa en ensamblador que, dadas las coordenadas cartesianas de sendos puntos pertenecientes a una función **f(x)**, calcule el valor de la función para un cierto punto intermedio de coordenada **x_p** dada.

A continuación se muestra una posible solución en C:

```
int x1, y1, x2, y2, xp, yp;
int main (void) {
    yp = ((xp-x1) * (y2-y1)) / (x2-x1) + y1;
    return 0;
}
```

Las variables **x1**, **x2**, **y1**, **y2**, **xp** e **yp** deben existir en la solución con el mismo nombre, y tendrán un valor inicial desconocido. Para realizar las pruebas se introducirán manualmente los valores necesarios en las variables. Escribir la solución en un fichero llamado **interpolacion_lineal.asm**.

3. Realizar un programa en ensamblador que evalúe la siguiente expresión con datos booleanos:

$$zb = (x \leq y) \parallel (xb \& yb)$$

Las variables **x** e **y** son números enteros con signo, mientras que **xb**, **yb** y **zb** son datos booleanos.

A continuación se muestra una posible solución en C (los datos booleanos aparecen como enteros):

```
int x, y, xb, yb, zb;
int main (void) {
    zb = (x<=y) || (xb & yb);
    return 0;
}
```

Las variables **x**, **y**, **xb**, **yb** y **zb** deben existir en la solución con el mismo nombre, y tendrán un valor inicial desconocido. Para realizar las pruebas se introducirán manualmente los valores necesarios en las variables. Escribir la solución en un fichero llamado **expresion_booleana.asm**.

4. Realizar un programa en ensamblador que, dado un número entero, lo cambie de signo por el método de invertir todos sus bits y sumar un 1 al resultado.

A continuación se muestra una posible solución en C:

```
int x, y;
int main (void) {
    y = (x ^ 0xFFFFFFFF) + 1;
    return 0;
}
```

Las variables **x** e **y** deben existir en la solución con el mismo nombre, y tendrán un valor inicial desconocido. Para realizar las pruebas se introducirá manualmente el valor necesario en la variable x. Escribir la solución en un fichero llamado **cambiar_signo.asm**.

5. Realizar un programa en ensamblador que, dados dos números enteros **x** y **lon** realice los siguientes desplazamientos de **x** con longitud **lon**:

- Desplazamiento lógico a la izquierda (resultado en la variable **xsll**).
- Desplazamiento lógico a la derecha (resultado en la variable **xsrl**).
- Desplazamiento aritmético a la derecha (resultado en la variable **xsra**).
- Rotación a la izquierda (resultado en la variable **xrol**).
- Rotación a la derecha (resultado en la variable **xror**).

Las variables **x**, **lon**, **xsll**, **xsrl**, **xsra**, **xrol** y **xror** deben existir en la solución con el mismo nombre, y tendrán un valor inicial desconocido. Para realizar las pruebas se introducirán manualmente los valores necesarios en las variables. Escribir la solución en un fichero llamado **desplazamientos.asm**.

A la vista de los resultados, identificar en qué casos estas operaciones corresponden con multiplicaciones o divisiones del dato original por una potencia de 2.