

Práctica 2. Estructuras de control en MIPS

1. Traduzca a lenguaje ensamblador de MIPS los siguientes fragmentos de código de lenguaje de alto nivel, considerando que todas las variables son enteras de 32 bits. Al comienzo de cada programa inserte instrucciones para dar diferentes valores a las variables, que se implementarán mediante registros (excepto los vectores que se implementarán en memoria). Compruebe los resultados de la ejecución con el simulador dando diferentes valores a esas variables.

a) `if (i>=0)`
 `f = i;`
 `else`
 `f = -i;`

b) `i=0;`
 `while (i<k)`
 `i++;`

c) `i=0;`
 `while (c[i]!=k)`
 `i++;`

d) `for (i=0; i<10;i++)`
 `c[i] = 4*i;`

2. Escriba un programa en lenguaje ensamblador de MIPS que deposite en el registro `$s5` la suma de los n primeros números naturales. Suponga que el valor n es el contenido del registro `$s3`.
3. Dados tres vectores de 10 elementos enteros de 32 bits cada uno, denominados respectivamente `a`, `b` y `c`, escriba un programa en MIPS que almacene en cada posición de `c` el resultado de sumar los valores de las posiciones correspondientes de los vectores `a` y `b`. En la declaración de variables inicialice los vectores `a` y `b` con valores cualesquiera.
4. A partir de la posición de memoria `A` se tiene almacenada una cadena de caracteres seguida de su terminador de cadena nulo. Escriba un programa en MIPS que copie esa cadena a partir de la posición de memoria `B`.
5. **(Entregable)** A partir de la posición de memoria `A` declare una cadena de caracteres cualquiera seguida de su terminador nulo. Escriba un programa en MIPS que copie a partir de la posición de memoria `B` los caracteres impares (primero, tercero, ...) y a partir de la posición de memoria `C` los caracteres pares (segundo, cuarto, ...). En ambos casos la copia se hará en orden inverso, es decir, empezando desde el último carácter de la cadena `A`. Además, en una posición de memoria `D` se almacenará el número de palabras contenidas en la cadena de caracteres inicial.

Ejemplo:

<code>A → "Hola Mundo\0"</code>
<code>B → "du lH\0"</code>
<code>C → "onMao\0"</code>
<code>D → 2</code>