Logotipo, nombre de la empresa

Descripción generada automáticamente

**Tarea N° 01**

**ESCUELA PROFESIONAL: Ciencia de la Computación**

**SEMESTRE : 2021-02**

**CURSO : Arquitectura de computadores**

**DOCENTE : Dr. Yván Jesús Túpac Valdivia**

**NOMBRES Y APELLIDOS: Fredy Goolberth Quispe Neira**

Texto

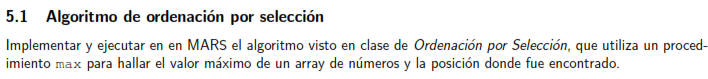
Descripción generada automáticamente

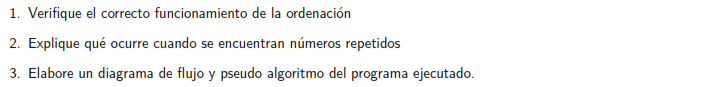
El código es distinto, debido a que éste posee en su área de código algunas pseudo-instrucciones como lo son “li $s1, 25; la $s1, variable” (load immediate and load address), éstos no son instrucciones de miniMIPS, son un conjunto de ellos; como en el caso del primero, éste en miniMIPS se puede hacer con una instrucción llamada “addi $s1, $zero, 25” el cual hace lo mismo en instrucciones que sí existen.

Texto

Descripción generada automáticamente con confianza media

Se entregará el informe de acuerdo con lo indicado en la plantilla de trabajo.







Texto

Descripción generada automáticamente

**INFORME:**

* Primeramente, empieza asignándole el valor 11 (lenght array) al registro $s1
* Le asigna al registro $s0 la dirección en memoria del array.
* Inicializa los registros $t0, $t2 y $t3 en 0.
* Inicia un bucle con una condición de pare, el cual es, si los registros $t2 y $s1 son iguales, finaliza el bucle. Las instrucciones dentro de él son:

1. Inicializa el registro $t4 asignándole la dirección que $s0 tiene del array más la variable en el registro $t3.
2. Carga en $t1 la variable de 32-bit que $t4 apunta.
3. Acumula en $t0 sumando el mismo con la variable que se cargó en $t1.
4. Aumenta en una unidad el valor del registro $t2, el cual se verifica en la condición de pare.
5. Asigna a $t3 el valor que $t2 tiene.
6. En las siguientes instrucciones suma 4 veces el valor que tenía $t3, esto con el fin de avanzar de 4 en 4 bytes, para obtener el siguiente elemento del array.
7. Se verifica la condición, si ésta cumple termina el bucle; caso contrario, la siguiente instrucción continúa el bucle enviando al PC al inicio del bucle.

* Después de que termine el bucle, todos los elementos están sumados en $t0, la siguiente instrucción copia el valor que éste contiene en $f8, el cual es un registro en punto flotante dentro del coproc1.
* Se hace lo mismo con $s1, el cual tiene la dimensión del array, se copia en $f9.
* Puesto que éstos datos vienen del procesador, éstos están en formato entero, se les hace la conversión a punto flotante teniendo como destino la misma dirección de memoria.
* Se hace la división en flotante de $f8/$f9 y se guarda el resultado en $f12. Esto para realizar la impresión de la variable flotante.
* Se hacen impresiones de cadenas.
* Finalmente se imprime el resultado y termina la ejecución del programa.