**Estructura de los computadores**

Practica 4 5 6

Francisco Joaquín Murcia Gómez 48734281H

Grupo 3

**Índice**

**Practica 4**………………………………………………………………………3-10

Aritmética de enteros (3) y funciones

Ejemplos …………………………………………………………………3-8

Entregas ………………………………………………………………...9-10

**Practica 5**…………………………………………………………….………11-14

Estructuras de control

Ejemplos ………………………………………...……………………11-12

Entregas …………………………………………………………...….13-14

**Practica 6**….…………………………………………………………………15-18

Variables

Ejemplos …………………………………………………………………...15

Entregas ………………………………………………………………16-18

**Practica 4**

**Ejemplos**

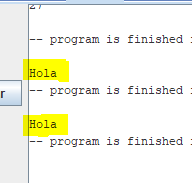
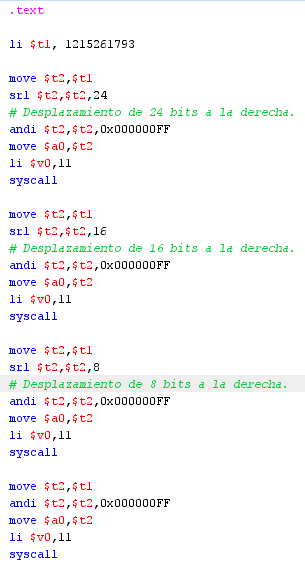
1. **Instrucciones de desplazamiento**
   1. **¿Con qué instrucciones traducirá el ensamblador las pseudoinstrucciones rol y ror?.**

“ori”, “sll”, “srl” y ”or”

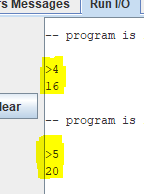
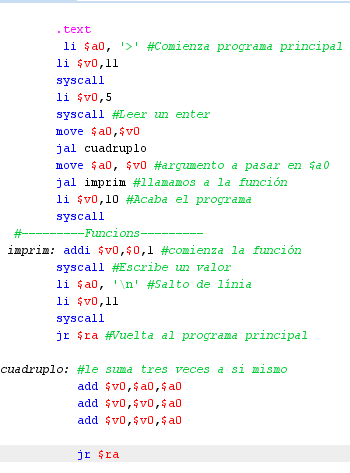
* 1. **Las instrucciones de desplazamiento siguen el formado tipo R. ¿Cuál es la codificación en hexadecimal de la instrucción ssl $t2,$t1,3? Y de la instrucción srl $t2,$t1,7. ¿Cuál es el valor de cada campo? Ayúdate ensamblado un código de prueba.**

0x000950c0 , 0x000951c2

* 1. **Descubre la palabra escondida. Dado el siguiente código, complétalo de tal manera que mediante instrucciones lógicas y de desplazamientos puedas escribir en la consola cada uno de los caracteres que se encuentran almacenados en cada byte del registre $t1.**



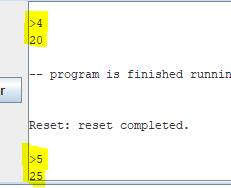
1. **Funciones del programa**
   1. **Escribe una función con instrucciones suma que devuelve el cuádruplo del número entero que se le pasa. Escribid el programa principal que lea el número del teclado y escriba el cuádruplo en la consola aprovechando la función imprim.**

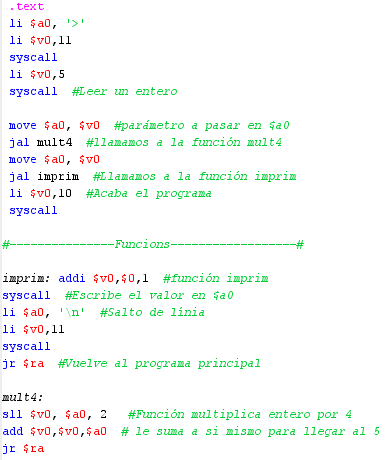


1. **Convenio de los registros del MIPS**
   1. **Observad el último código escrito, ¿se ajusta al convenio de MIPS de utilización de registros? De ahora en adelante haced servir el convenio.**

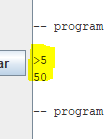
No, porque no utiliza el $s

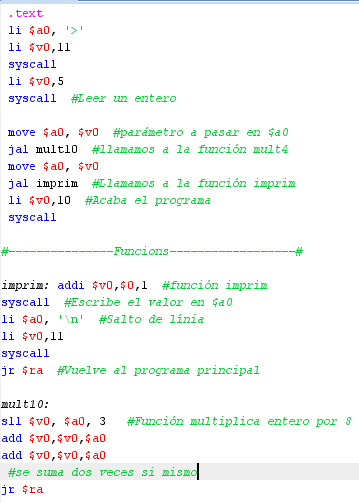
1. **Multiplicación por desplazamientos y sumas**
   1. **Modifica el código en el que ahora haya una función multi5 que multiplique por 5 y muestre el resultado por consola. Comprobar que el resultado es correcto.**



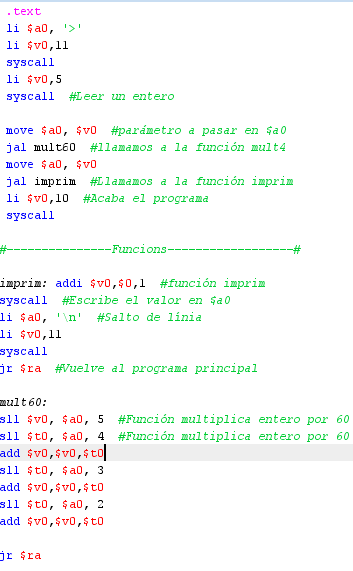
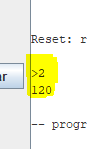


* 1. **Modifícalo ahora para tener una nueva función mult10 que multiplique por 10. Comprobar que el resultado es correcto.**

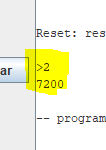


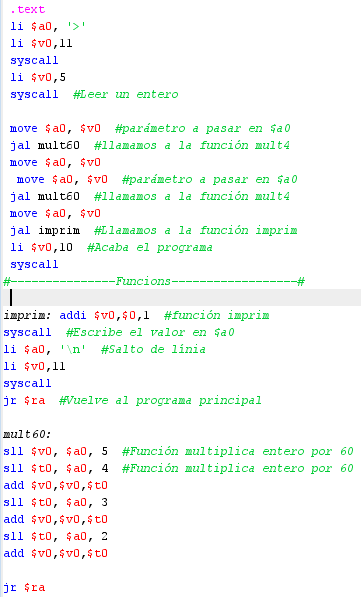


* 1. **Escribe una función que multiplique por 60. Escribe el programa principal que lea una cantidad de minutos y devuelve por consola la cantidad en segundos.**

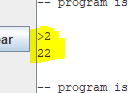


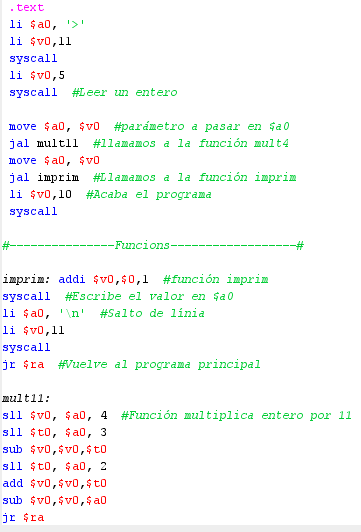
* 1. **Modifica el código de tal manera que ahora lo que lea sea una cantidad de hora y muestre por consola la cantidad de segundos.**





1. **Multiplicación por desplazamientos, sumas y restas**
   1. **Escribe la función que multiplique por la constante 11 según el algoritmo de Booth y comprueba que el resultado es correcto.**

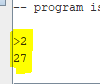




**Entregas**

**Escribe el código que lee el valor x y escribe por pantalla la solución de la ecuación: 5x^2+2x+3.**

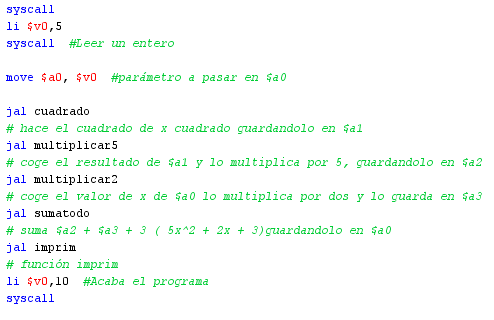
Pruebas:

 Si x=2 🡪 resultado 27

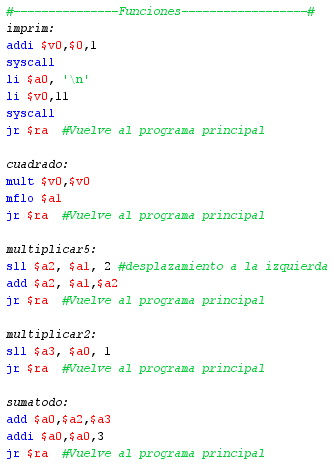
Si x=5 🡪 resultado 138

Si x=10 🡪 resultado 523

Main:



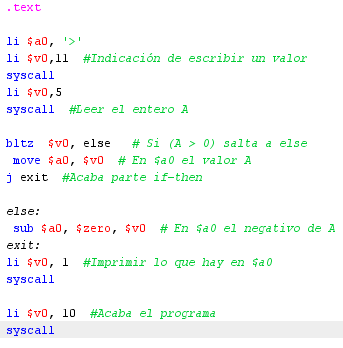
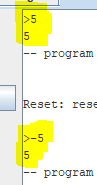
Funciones:

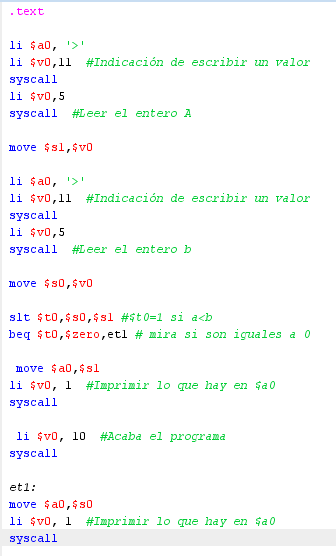


**Practica 5**

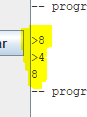
**Ejemplos**

1. **Como hacer uso de las instrucciones de salto condicionales**
   1. **La instrucción bltz (branch if less than zero) salta si el valor es menor que cero y es similar a bgez ya que también compara un registro con 0 pero siendo contraria la condición de salto. Cambiad la instrucción bgez por bltz, ¿Qué modificaciones tendríais que hacer en el código?**

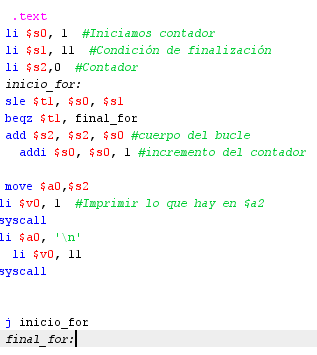
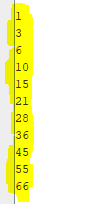
 

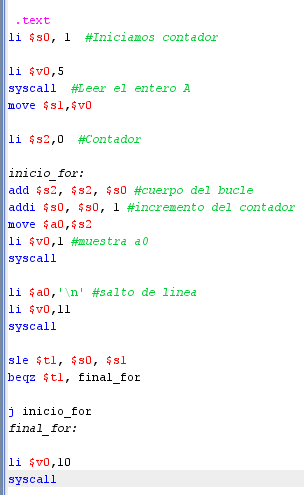
* 1. **Escribid el programa que lea dos enteros del teclado y escriba en la consola el mayor.**

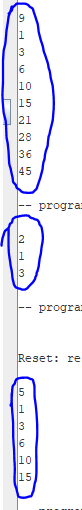




1. **Bucles**

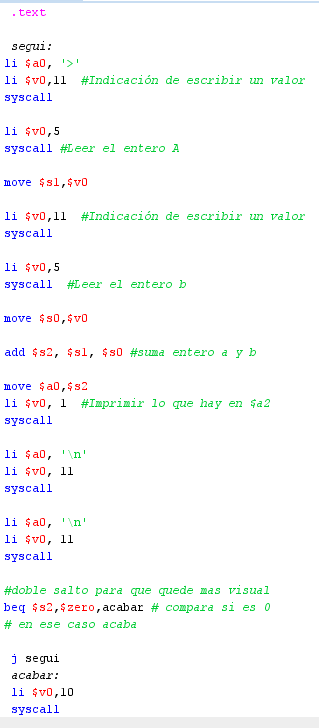
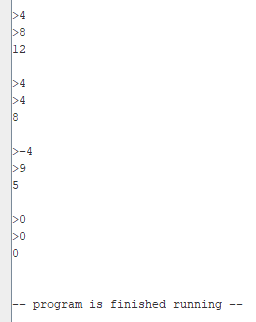
 

**Añade al código anterior la posibilidad de leer un valor n por teclado y escribe el resultado de la suma por consola. Haz que haya un bucle infinito que lea por teclado excepto en el caso de que se escriba 0, en el cual se saldrá del programa.**

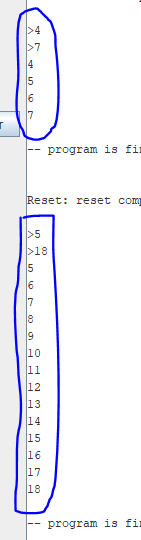
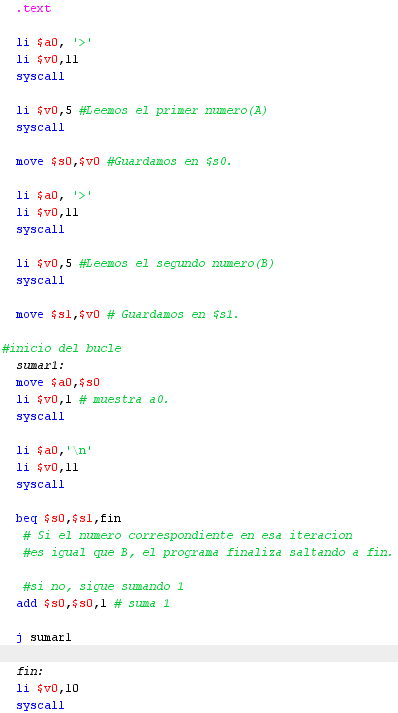


**Entregas**

1. **Haz el código que lea dos enteros de la consola y escriba la suma y vuelva a comenzar si el resultado es distinto de 0. Es pseudocódigo sería:**



1. **Haz el código que lee de teclado dos valores positivos A y B en los que A <B. El programa tiene que escribir por consola los valores comprendidos entre ambos, incluyéndolos a ellos mismos. Es decir, si A=3 y B=6, escribe en la consola 3 4 5 6 (puedes escribir, por ejemplo, un salto de línea después de cada uno de los valores a mostrar).**

**Practica 6**

**Ejemplos**

1. **Acceso a la memoria**
   1. **¿Cuántos bytes de la memoria principal están ocupados por datos del programa?**

12 bytes

* 1. **¿Cuántas instrucciones de acceso a la memoria contiene el programa?**

3

* 1. **¿Qué valor tiene el registro $t1 cuando se ejecuta la instrucción lw $s1,0($t1)?**

La direccion de b

* 1. **¿En qué dirección se almacena el resultado?**

En c

* 1. **Sustituid la instrucción sw $s2,0($t2) por sw $s2,2($t2) ¿Qué ocurre cuando se intenta ejecutar el programa? Razonad la respuesta**

Error, cogemos una palabra que no es multiplo de 4

* 1. **¿Cuál es la codificación en lenguaje máquina de la instrucción lw $s1,0($t1)? Desglosa la instrucción en los distintos campos del formato.**

0x8d310000🡪8d=10001101🡪 1001

1. **Uso de la memoria**
   1. **¿Cuál es la dirección del byte donde está el carácter ‘1’?**

0x10010020 + 4=0x10010024

1. **Los punteros**
   1. **Ensambla el código y ejecútalo. ¿Qué hace la función 4 para la instrucción syscall?**

Imprime cadenas de caracters

* 1. **¿Cuál es la codificación máquina de la instrucción syscall?**

0x000000c

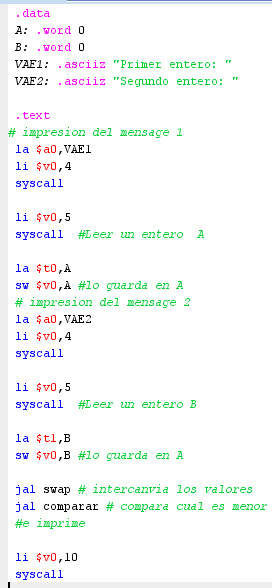
* 1. **¿En qué dirección se guarda el resultado de la suma?**

0x10010008

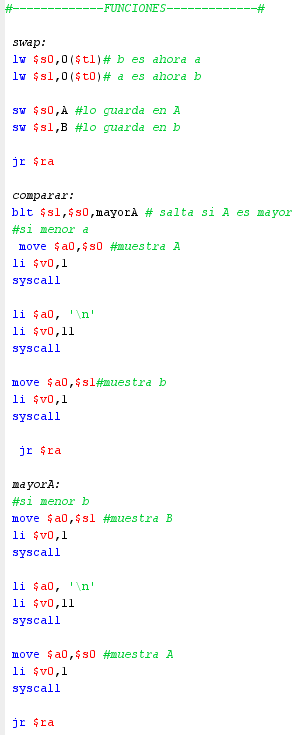
**Entregas**

**Escribe el código que lee dos enteros del teclado mostrando sendos mensajes por consola: uno que pida al usuario que introduzca el primer valor y tras haberlo leído que muestre otro solicitando el segundo valor. Los datos se almacenarán en la memoria, para lo cual debes haber reservado previamente espacio en el segmento de datos. Una vez almacenados los datos tienes que llamar a una función, que denominaremos SWAP, que intercambie el contenido de las dos posiciones de memoria. Para finalizar se leerán los valores guardados en la memoria y se mostrarán ordenados de menor a mayor en la pantalla.**

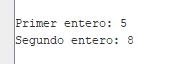
MAIN:



FUNCIONES:



Ejemplo de ejecucion:

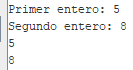
se introducen los datos

se gurdan en memoria



los intercanvia



finalmente los ordena e inprime

