



Universidad La Salle Arequipa
Ingeniería de Software
Fundamentos de Lenguajes de Programación

Práctica 3

Ensamblador

Hecho por: Yoshiro Milton Miranda Valdivia

Docente: Vicente Enrique Machaca Arceda

https://github.com/ymirandav/Fundamentos_de_Lenguaje_de_programacion

8 de septiembre de 2021

Ejercicio 1

1. Implementar un programa que muestre la suma, la diferencia, la multiplicación, la división y el promedio de dos números ingresados por teclado.
-

Respuesta

```
.data
lin1: .ascii "Ingresa un numero: "
lin2: .ascii "Ingresa otro numero: "
lin3: .ascii "\nLa suma es: "
lin4: .ascii "\nLa resta es: "
lin5: .ascii "\nLa multiplicacion es: "
lin6: .ascii "\nLa division es: "
lin7: .ascii "\nEl promedio es: "

.text
main:
#-----
    li    $v0, 4
    la    $a0, lin1
    syscall

    li    $v0, 5
    syscall
    move  $t0, $v0

    li    $v0, 4
    la    $a0, lin2
    syscall

    li    $v0, 5
    syscall
    move  $t1, $v0
#-----

#Suma
    add   $t2, $t0, $t1

#Diferencia
```

```
#Multiplicacion
    mul   $t4, $t0, $t1

#Division
    mtc1   $t0, $f1
    cvt.s.w $f1, $f1

    mtc1   $t1, $f2
    cvt.s.w $f2, $f2

exit:

#Mostrar suma
    li    $v0, 4
    la    $a0, lin3
    syscall

    move  $a0, $t2
    li    $v0, 1
    syscall

#Mostrar diferencia
    li    $v0, 4
    la    $a0, lin4
    syscall

    move  $a0, $t3
    li    $v0, 1
    syscall
```

```

#Mostrar multiplicacion
    li    $v0,4
    la    $a0,lin5
    syscall


    move  $a0, $t4
    li    $v0, 1
    syscall

#Mostrar division
    li    $v0,4
    la    $a0,lin6
    syscall

    div.s  $f12,$f1,$f2
    li    $v0, 2
    syscall

#fin
    mov.s $f8, $f1
    li    $v0,10
    syscall

```

 Console

```

Ingresa un numero: 3
Ingresa otro numero: 2

La suma es: 5
La resta es: 1
La multiplicacion es: 6
La division es: 1.50000000|

```

Ejercicio 2

2. Implementar un programa que solicite una cantidad n de números y luego retorne: la suma de estos, el promedio, el mayor y el menor.
-

Respuesta

```
.data
ite: .asciiz "Ingrese la cantidad de numeros: "
num: .asciiz "Ingrese un numero: "
suma: .asciiz "La suma es: "
prom: .asciiz "\nEl promedio es: "
mayor: .asciiz "\nEl mayor es: "
menor: .asciiz "\nEl menor es: "

.text
main:

    li $v0, 4
    la $a0, ite
    syscall

    li $v0, 5
    syscall
    move $t0, $v0

    li $t1, 0 #index
    li $t4, 0
    li $t7, 10000000000000000
    li.s $f1, 1.0

loop:
    beq $t1, $t0, exit

    li $v0, 4
    la $a0, num
    syscall
```

```
li $v0, 5
syscall
move $t2, $v0
move $t6, $v0
move $t9, $v0

mtc1 $t9, $f2
cvt.s.w $f2, $f2

add $t3, $t3, $t2

add.s $f12, $f12, $f2
add.s $f3, $f3, $f1

bge $t4, $t2, if
else:
    move $t5, $t2
    move $t4, $t2
if:
    move $t5, $t4
end_if:

ble $t7, $t6, if2
else2:
    move $t8, $t6
    move $t7, $t6
if2:
    move $t8, $t7
end_if2:

add $t1, $t1, 1
j loop
```

```

exit:

#-----SUMA
li $v0, 4
la $a0, suma
syscall

li $v0, 1
move $a0, $t3
syscall

#-----MAYOR
li $v0, 4
la $a0, mayor
syscall

li $v0, 1
move $a0, $t5
syscall

#-----MENOR
li $v0, 4
la $a0, menor
syscall

li $v0, 1
move $a0, $t8
syscall

jr $ra

```

Console

```

Ingrese la cantidad de numeros: 4
Ingrese un numero: 3
Ingrese un numero: 2
Ingrese un numero: 2
Ingrese un numero: 3
La suma es: 10
El mayor es: 3
El menor es: 2|

```

Ejercicio 3

3. Implemente un programa que solicite por teclado: la longitud de los tres lados de un triangulo. Luego el programa debe indicar si es un triangulo valido.
-

Respuesta

```
.data
    st1: .asciiz "Ingrese el primer lado:\n"
    st2: .asciiz "Ingrese el segundo lado:\n"
    st3: .asciiz "Ingrese el tercer lado:\n"
    st4: .asciiz "El triangulo es valido\n"
    st5: .asciiz "El triangulo no es valido\n"
.text
main:

#-----

    la $a0, st1
    li $v0, 4
    syscall

    li $v0, 5
    syscall
    move $t1, $v0

    la $a0, st2
    li $v0, 4
    syscall

    li $v0, 5
    syscall
    move $t2, $v0

    la $a0, st3
    li $v0, 4
    syscall

    li $v0, 5
    syscall
    move $t3, $v0

    add $a0, $t1, $t2
    move $t4, $a0
    #li $v0, 1
    #syscall

    add $a0, $t1, $t3
    move $t5, $a0
    #li $v0, 1
    #syscall

    add $a0, $t2, $t3
    move $t6, $a0
    #li $v0, 1
    #syscall

#-----
```

```

#-----

bge $t3, $t4 LABEL_IF
LABEL_ELSE:

    bge $t2, $t5 bLABEL_IF

    bLABEL_ELSE:

        bge $t1, $t6 cLABEL_IF

        cLABEL_ELSE:

            la $a0, st4
            b END_LABEL_IF
        cLABEL_IF:
            la $a0, st5

    bLABEL_IF:

        la $a0, st5

#-----

LABEL_IF:
    la $a0, st5

END_LABEL_IF:
    li $v0, 4
    syscall

jr $ra

```

Console

```

Ingrese el primer lado:
2
Ingrese el segundo lado:
3
Ingrese el tercer lado:
2
El triangulo es valido
|

```