

## Código de Assembler



- **Alumno:** Jeriel Estrada Candiano
- **Profesor:** Santiago Trini
- **Materia:** CATE
- **Fecha de Entrega:** 14/10/2025
- **Ciclo Lectivo:** 2025

### **Código:**

```
.data
newline: .asciiz "\n"
```

```
.text
.globl main
```

fill:

```
addi $sp, $sp, -16
sw   $ra, 12($sp)
sw   $s0, 8($sp)
```

```
move $s0, $a0
move $s1, $a1
move $s2, $a2
```

```
li   $t0, 0      # i = 0
```

fill\_loop:

```
beq  $t0, $s1, fill_done
```

```
sll  $t1, $t0, 2
add  $t2, $s0, $t1
```

```
sw   $s2, 0($t2)
```

```
addi $s2, $s2, 1
addi $t0, $t0, 1
j    fill_loop
```

fill\_done:

```
lw   $s0, 8($sp)
lw   $ra, 12($sp)
addi $sp, $sp, 16
jr   $ra
```

max:

```
addi $sp, $sp, -24
```

```
sw    $ra, 20($sp)
sw    $s0, 16($sp)
sw    $s1, 12($sp)
```

```
move  $s0, $a0
move  $s1, $a1
```

```
lui   $t0, 0x8000
addi  $t0, $t0, 0
move  $t1, $t0
```

```
li    $t2, 0      # i = 0
```

max\_loop:

```
beq   $t2, $s1, max_done
```

```
sll   $t3, $t2, 2
add   $t4, $s0, $t3
lw    $t5, 0($t4)
```

```
slt   $t6, $t1, $t5
beq   $t6, $zero, skip_update
move  $t1, $t5
```

skip\_update:

```
addi  $t2, $t2, 1
j     max_loop
```

max\_done:

```
move  $v0, $t1
```

```
lw    $s1, 12($sp)
lw    $s0, 16($sp)
lw    $ra, 20($sp)
addi  $sp, $sp, 24
jr    $ra
```

main:

```
addi  $sp, $sp, -432
```

```
sw    $ra, 428($sp)
sw    $fp, 424($sp)
addi  $fp, $sp, 432
```

```
addi  $t0, $fp, -400
```

```
move  $a0, $t0
li    $a1, 100
li    $a2, 87
jal   fill
```

```
move  $a0, $t0
li    $a1, 100
jal   max
```

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

```
la    $a0, newline
li    $v0, 4
syscall
lw    $ra, 428($sp)
lw    $fp, 424($sp)
addi  $sp, $sp, 432
```

```
li    $v0, 10
syscall
```

### **Instrucciones en Binario:**

Instrucción: add \$t2, \$s0, \$t1 / Binario:

00000010000010010101000000100000

Instrucción: addi \$t0, \$t0, 1/ Binario:

00100001000010000000000000000001

Instrucción: lw \$t1, 0(\$t0) / Binario:

10001101000010010000000000000000