

Isac Martins

Lab 8


```
1 .text
2 li $t1, 1
3 li $v0, 5
4 syscall
5 move $t0, $v0
6 move $t8, $t0
7 jal factorial
8 move $s1, $s0 ## N!
9 li $v0, 5
10 syscall
11 move $t0, $v0
12 move $t9, $t0
13 jal factorial
14 move $s2, $s0 ## P!
15 sub $t8, $t8, $t9
16 move $t0, $t8
17 jal factorial
18 move $s3, $s0 ## (n- P)!
19 mul $s4, $s2, $s3
20 div $s4, $s1, $s4
21 j End
22 ## Soma
23 retornoFactorial:
24 move $s0, $t1
25 li $t1, 1
26 jr $ra
27 factorial:
28 beq $t0, $zero, retornoFactorial
```

Nesse código uso de um procedimento retornando usando uma entrada em um registrador genérico e a saída em um registrador genérico, assim posso chamar a função múltiplas vezes com um código que reaproveita as entradas


Resultados Programa

```
4
2
6
-- program is finished runn
8
5
56
-- program is finished runn
```

Resultados Calculadora


$$(4!) / ((2!) * (2!)) =$$

6


$$(8!) / ((5!) * (3!)) =$$

56