

2023/1 CCT ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

[Painel](#) / [Meus cursos](#) / [Departamento de Ciência da Computação](#) / [Bacharelado em Ciência da Computação](#) / [2023/1 CC](#)
/ [2023/1 CCT CCI192-04U AOC0004](#) / [Semana 05](#) / [Q03 - Programação MIPS - Fibonacci recursivo](#)


 [Descrição](#)

 [Visualizar envios](#)

Nota



Revisado em Saturday, 8 Apr 2023, 19:05 por

Nota: 10,00 / 10,00

Redução final: 0 [2 / 3 -1] 

Relatório de avaliação  [-]

[\[+\]](#) [Summary of tests](#)

 Enviado em Saturday, 8 Apr 2023, 19:05  [Baixar](#)

fibonacciRecursivo.s

```
1  # autor: Kauan Henrique Werlich
2  # disciplina: Arquitetura e Organização de Computadores (AOC)
3  .text
4  .globl main
5  main:
6      li $v0, 5 # ler inteiro
7      syscall
8      add $a0, $v0, $zero # $a0 = $v0
9      jal fibonacci_Rec
10
11     move $a0, $v0 # colocar a saída da função para ser impressa no console
12     li $v0, 1 # imprimir inteiro
13     syscall
14 end:
15     li $v0, 10 # código para terminar o programa
16     syscall
17
18 fibonacci_Rec:
19     slti $t0, $a0, 2
20     bne $t0, $zero, fator_parada # se $a0 for diferente de 0 (igual a 1) vá para o branch
21
22     # salvar contexto
23     addi $sp, $sp, -12 # deslocando o topo da pilha 12 bytes
24     sw $s0, 0($sp) # armazenando $s0 nos últimos 4 bytes da pilha
25     sw $s1, 4($sp) # armazenando $s1 nos 4 bytes após $s0
26     sw $ra, 8($sp) # armazenando $ra nos 4 bytes após $s1
27
28     #código principal da função recursiva
29     move $s1, $a0
30     addi $a0, $a0, -1
31     jal fibonacci_Rec
32     move $s0, $v0
33
34     addi $a0, $s1, -2
35     jal fibonacci_Rec
36     move $s1, $v0
37
38     add $v0, $s0, $s1
39
40     # restaurar
41     lw $s0, 0($sp) # restaurando o valor de $s0
42     lw $s1, 4($sp) # restaurando o valor de $s1
43     lw $ra, 8($sp) # restaurando o valor de $ra
44     addi $sp, $sp, 12 # ajustando topo da pilha para "excluir" os itens
45
46     jr $ra #retornar
47
48 fator_parada: # se a entrada for menor que 2, a função retorna 1
49     addi $v0, $zero, 1
50     jr $ra
51
```



[VPL](#)

Você acessou como KAUAN HENRIQUE WERLICH (Sair)
2023/1_CCT_CCI192-04U_AOC0004

Português - Brasil (pt_br)

Deutsch (de)

English (en)

Español - Internacional (es)

Français (fr)

Italiano (it)

Português - Brasil (pt_br)

日本語 (ja)

Resumo de retenção de dados

Obter o aplicativo para dispositivos móveis

