

Nomes: Phillip Corrêa Mittmann e Robson Felipe Bittencourt

Programas

Todos os programas foram feitos, porém apenas o primeiro funciona de maneira devida, os outros três apresentam defeitos. Não foram alteradas instruções.

- P1 - 10 primeiros números da sequência de Fibonacci.
- P2 - Escrever N números da sequência de Fibonacci a partir de um valor na memória.
 - Escreve números de maneira crescente.
- P3 - Escrever o fatorial de um valor pré determinado na memória.
 - O resultado escrito na memória é sempre 1.
- P4 - Bubble sort.
 - Não ordena os valores.

VM

A VM foi criada na linguagem Java. É necessário ter gradle previamente instalado na máquina para utilizá-la.

Para executar os testes, é necessário descomentar o programa desejado no arquivo App.java, conforme abaixo.

```
public static void main(String[] args) {  
  
    vm = new VM();  
  
    vm.p1();  
    // vm.p2();  
    // vm.p3();  
    // vm.p4();  
}
```

Resultados

- P1

```
----- programa carregado
0: [ LDI, 0, -1, 0 ]
1: [ STD, 0, -1, 20 ]
2: [ LDI, 1, -1, 1 ]
3: [ STD, 1, -1, 21 ]
4: [ LDI, 7, -1, 22 ]
5: [ LDI, 5, -1, 6 ]
6: [ LDI, 6, -1, 31 ]
7: [ LDI, 2, -1, 0 ]
8: [ ADD, 2, 0, -1 ]
9: [ LDI, 0, -1, 0 ]
10: [ ADD, 0, 1, -1 ]
11: [ ADD, 1, 2, -1 ]
12: [ STX, 7, 1, -1 ]
13: [ ADDI, 7, -1, 1 ]
14: [ SUB, 6, 7, -1 ]
----- após execucao
Interrupcao intSTOP
0: [ LDI, 0, -1, 0 ]
1: [ STD, 0, -1, 20 ]
2: [ LDI, 1, -1, 1 ]
3: [ STD, 1, -1, 21 ]
4: [ LDI, 7, -1, 22 ]
5: [ LDI, 5, -1, 6 ]
6: [ LDI, 6, -1, 31 ]
7: [ LDI, 2, -1, 0 ]
8: [ ADD, 2, 0, -1 ]
9: [ LDI, 0, -1, 0 ]
10: [ ADD, 0, 1, -1 ]
11: [ ADD, 1, 2, -1 ]
12: [ STX, 7, 1, -1 ]
13: [ ADDI, 7, -1, 1 ]
14: [ SUB, 6, 7, -1 ]
15: [ JMPIG, 5, 6, -1 ]
16: [ STOP, -1, -1, -1 ]
17: [ ___, -1, -1, -1 ]
18: [ ___, -1, -1, -1 ]
19: [ ___, -1, -1, -1 ]
20: [ DADO, -1, -1, 0 ]
21: [ DADO, -1, -1, 1 ]
22: [ DADO, -1, -1, 1 ]
23: [ DADO, -1, -1, 2 ]
24: [ DADO, -1, -1, 3 ]
25: [ DADO, -1, -1, 5 ]
26: [ DADO, -1, -1, 8 ]
27: [ DADO, -1, -1, 13 ]
28: [ DADO, -1, -1, 21 ]
29: [ DADO, -1, -1, 34 ]
30: [ DADO, -1, -1, 55 ]
31: [ ___, -1, -1, -1 ]
```

- P2

```
----- programa carregado
0: [ LDI, 0, -1, -1 ]
1: [ LDI, 1, -1, 0 ]
2: [ LDI, 2, -1, 1 ]
3: [ LDI, 3, -1, 50 ]
4: [ LDX, 4, 3, -1 ]
5: [ LDI, 5, -1, 18 ]
6: [ LDI, 6, -1, 32 ]
7: [ JMPIL, 5, 4, -1 ]
8: [ JMPIE, 6, 4, -1 ]
9: [ STX, 3, 1, -1 ]
10: [ ADDI, 3, -1, 1 ]
11: [ SUBI, 4, -1, 1 ]
12: [ JMPIE, 6, 4, -1 ]
13: [ JMP, -1, -1, 20 ]
14: [ STX, 3, 0, -1 ]
----- após execucao
```

```
Interrupcao intSTOP
0: [ LDI, 0, -1, -1 ]
1: [ LDI, 1, -1, 0 ]
2: [ LDI, 2, -1, 1 ]
3: [ LDI, 3, -1, 50 ]
4: [ LDX, 4, 3, -1 ]
5: [ LDI, 5, -1, 18 ]
6: [ LDI, 6, -1, 32 ]
7: [ JMPIL, 5, 4, -1 ]
8: [ JMPIE, 6, 4, -1 ]
9: [ STX, 3, 1, -1 ]
10: [ ADDI, 3, -1, 1 ]
11: [ SUBI, 4, -1, 1 ]
12: [ JMPIE, 6, 4, -1 ]
13: [ JMP, -1, -1, 20 ]
14: [ STX, 3, 0, -1 ]
15: [ JMP, -1, -1, 32 ]
16: [ LDI, 0, -1, 0 ]
17: [ LDI, 1, -1, 1 ]
18: [ LDI, 5, -1, 23 ]
19: [ LDI, 2, -1, 0 ]
20: [ ADD, 2, 1, -1 ]
21: [ LDI, 0, -1, 0 ]
22: [ ADD, 0, 1, -1 ]
23: [ ADD, 1, 2, -1 ]
24: [ STX, 3, 1, -1 ]
25: [ ADDI, 3, -1, 1 ]
26: [ SUBI, 4, -1, 1 ]
27: [ JMPIG, 5, 4, -1 ]
28: [ STOP, -1, -1, -1 ]
29: [ ___, -1, -1, -1 ]
30: [ ___, -1, -1, -1 ]
31: [ ___, -1, -1, -1 ]
32: [ ___, -1, -1, -1 ]
33: [ ___, -1, -1, -1 ]
34: [ ___, -1, -1, -1 ]
35: [ ___, -1, -1, -1 ]
36: [ ___, -1, -1, -1 ]
37: [ ___, -1, -1, -1 ]
```

```
38: [ ___, -1, -1, -1 ]
39: [ ___, -1, -1, -1 ]
40: [ ___, -1, -1, -1 ]
41: [ ___, -1, -1, -1 ]
42: [ ___, -1, -1, -1 ]
43: [ ___, -1, -1, -1 ]
44: [ ___, -1, -1, -1 ]
45: [ ___, -1, -1, -1 ]
46: [ ___, -1, -1, -1 ]
47: [ ___, -1, -1, -1 ]
48: [ ___, -1, -1, -1 ]
49: [ ___, -1, -1, -1 ]
50: [ DADO, -1, -1, 0 ]
51: [ DADO, -1, -1, 1 ]
52: [ DADO, -1, -1, 2 ]
53: [ DADO, -1, -1, 3 ]
54: [ DADO, -1, -1, 4 ]
55: [ DADO, -1, -1, 5 ]
56: [ DADO, -1, -1, 6 ]
57: [ DADO, -1, -1, 7 ]
58: [ DADO, -1, -1, 8 ]
59: [ DADO, -1, -1, 9 ]
60: [ DADO, -1, -1, 10 ]
61: [ DADO, -1, -1, 11 ]
```

- P3

```

----- programa carregado
0: [ DADO, 50, -1, 10 ]
1: [ LDI, 0, -1, -1 ]
2: [ LDI, 1, -1, 1 ]
3: [ LDI, 2, -1, 50 ]
4: [ LDX, 3, 2, -1 ]
5: [ LDI, 4, -1, 11 ]
6: [ LDI, 5, -1, 13 ]
7: [ LDI, 6, -1, 25 ]
8: [ JMPIL, 4, 3, -1 ]
9: [ JMPIE, 5, 3, -1 ]
10: [ JMP, -1, -1, 15 ]
11: [ STX, 2, 0, -1 ]
12: [ JMP, -1, -1, 24 ]
13: [ STX, 2, 1, -1 ]
14: [ JMP, -1, -1, 24 ]
----- após execucao
Interrupcao intSTOP
0: [ DADO, 50, -1, 10 ]
1: [ LDI, 0, -1, -1 ]
2: [ LDI, 1, -1, 1 ]
3: [ LDI, 2, -1, 50 ]
4: [ LDX, 3, 2, -1 ]
5: [ LDI, 4, -1, 11 ]
6: [ LDI, 5, -1, 13 ]
7: [ LDI, 6, -1, 25 ]
8: [ JMPIL, 4, 3, -1 ]
9: [ JMPIE, 5, 3, -1 ]
10: [ JMP, -1, -1, 15 ]
11: [ STX, 2, 0, -1 ]
12: [ JMP, -1, -1, 24 ]
13: [ STX, 2, 1, -1 ]
14: [ JMP, -1, -1, 24 ]
15: [ LDI, 4, -1, 0 ]
16: [ ADD, 4, 3, -1 ]
17: [ SUBI, 3, -1, 1 ]
18: [ STX, 2, 1, -1 ]
19: [ JMPIE, 6, 3, -1 ]
20: [ LDI, 0, -1, 21 ]
21: [ MULT, 4, 3, -1 ]
22: [ STX, 2, 4, -1 ]
23: [ SUB, 3, -1, 1 ]
24: [ JMPIG, 0, 3, -1 ]
25: [ STOP, -1, -1, -1 ]
26: [ ___, -1, -1, -1 ]
27: [ ___, -1, -1, -1 ]
28: [ ___, -1, -1, -1 ]
29: [ ___, -1, -1, -1 ]
30: [ ___, -1, -1, -1 ]
31: [ ___, -1, -1, -1 ]
32: [ ___, -1, -1, -1 ]
33: [ ___, -1, -1, -1 ]
34: [ ___, -1, -1, -1 ]
35: [ ___, -1, -1, -1 ]
36: [ ___, -1, -1, -1 ]
37: [ ___, -1, -1, -1 ]
38: [ ___, -1, -1, -1 ]
39: [ ___, -1, -1, -1 ]
40: [ ___, -1, -1, -1 ]
41: [ ___, -1, -1, -1 ]
42: [ ___, -1, -1, -1 ]
43: [ ___, -1, -1, -1 ]
44: [ ___, -1, -1, -1 ]
45: [ ___, -1, -1, -1 ]
46: [ ___, -1, -1, -1 ]
47: [ ___, -1, -1, -1 ]
48: [ ___, -1, -1, -1 ]
49: [ ___, -1, -1, -1 ]
50: [ DADO, -1, -1, 1 ]
51: [ ___, -1, -1, -1 ]
52: [ ___, -1, -1, -1 ]
53: [ ___, -1, -1, -1 ]
54: [ ___, -1, -1, -1 ]
55: [ ___, -1, -1, -1 ]
56: [ ___, -1, -1, -1 ]
57: [ ___, -1, -1, -1 ]
58: [ ___, -1, -1, -1 ]
59: [ ___, -1, -1, -1 ]
60: [ ___, -1, -1, -1 ]
61: [ ___, -1, -1, -1 ]

```

- P4

```

----- programa carregado
0: [ DADO, 0, 0, 5 ]
1: [ DADO, 1, 0, 3 ]
2: [ DADO, 2, 0, 4 ]
3: [ DADO, 3, 0, 1 ]
4: [ DADO, 4, 0, 2 ]
5: [ LDI, 3, -1, 6 ]
6: [ LDI, 2, -1, 5 ]
7: [ LDI, 0, -1, 1 ]
8: [ LDI, 1, -1, 1 ]
9: [ JMPIL, 4, 2, -1 ]
10: [ STOP, -1, -1, -1 ]
11: [ LDX, 3, 0, -1 ]
12: [ LDX, 4, 1, -1 ]
13: [ SUB, 3, 4, -1 ]
14: [ LDI, 4, -1, 12 ]
----- após execucao
Interrupcao intSTOP
0: [ DADO, 0, 0, 5 ]
1: [ DADO, 1, 0, 3 ]
2: [ DADO, 2, 0, 4 ]
3: [ DADO, 3, 0, 1 ]
4: [ DADO, 4, 0, 2 ]
5: [ LDI, 3, -1, 6 ]
6: [ LDI, 2, -1, 5 ]
7: [ LDI, 0, -1, 1 ]
8: [ LDI, 1, -1, 1 ]
9: [ JMPIL, 4, 2, -1 ]
10: [ STOP, -1, -1, -1 ]
11: [ LDX, 3, 0, -1 ]
12: [ LDX, 4, 1, -1 ]
13: [ SUB, 3, 4, -1 ]
14: [ LDI, 4, -1, 12 ]
15: [ JMPIG, 4, 3, -1 ]
16: [ JMP, -1, -1, 4 ]
17: [ LDX, 3, 1, -1 ]
18: [ SWAP, 3, 4, -1 ]
19: [ STX, 0, 3, -1 ]
20: [ STX, 1, 4, -1 ]
21: [ ADDI, 0, -1, 1 ]
22: [ ADDI, 1, -1, 1 ]
23: [ JMP, -1, -1, 4 ]

```

Anexos

Link com o código atualizado -

https://github.com/Fizoratti/jav_PUCRS-SisOp-Maquina-Virtual