**Universidade Federal de Alagoas - UFAL**
**Instituto de Computação - IC**
**Curso de Ciência da Computação**

**Professor Alcino Dall'Igna Júnior**

# Programas Exemplo

**Thalyssa de Almeida Monteiro**

**Maceió, 03 de dezembro de 2022**

## 1 Hello, World!

### 1.1 Programa

```
Begin {
    Show("Hello, World!");
}
```

### 1.2 Resultado

```
1 - Begin {

              [    2,     1] (    6, RW_BEGIN             ) Begin
              [    2,     2] (   48, DELI_OCURLY          ) {
2 -     Show("Hello, World!");

              [    3,     1] (   15, RW_SHOW              ) Show
              [    3,     2] (   44, DELI_OPAREN          ) (
              [    3,     3] (   40, IDEN_STRING          ) "Hello, World!"
              [    3,     4] (   45, DELI_CPAREN          ) )
              [    3,     5] (   51, DELI_SEMICOL         ) ;
3 - }
              [    4,     1] (   49, DELI_CCURLY          ) }
              [    4,     1] (   57, OTHER_EOF            ) EOF
```

## 2 Fibonacci

### 2.1 Programa

```
Begin {
    Function Int fibonacci(Int value) {
        Int last = 1;
        Int penultimate = 1;
        Int next;

        If value == 1 {
            Return 1;
        }
        If value == 2 {
            Return 1;
        }

        Int count = 3;
        While count <= value {
            next = last + penultimate;
            penultimate = last;
            last = next;
            count = count + 1;
        }

        Return next;
    }
    Int nthTerm;
    Int result;

    Get(nthTerm);

    result = fibonacci(value);
    Show("%d", result);
}
```

## 2.2 Resultado

```
1 - Begin {

              [   2,    1] (   6, RW_BEGIN             ) Begin
              [   2,    2] (  48, DELI_OCURLY          ) {
2 -     Function Int fibonacci(Int value) {

              [   3,    1] (  17, RW_FUNCTION          ) Function
              [   3,    2] (   1, RW_INT               ) Int
              [   3,    3] (  35, ID                   ) fibonacci
              [   3,    4] (  44, DELI_OPAREN          ) (
              [   3,    5] (   1, RW_INT               ) Int
              [   3,    6] (  35, ID                   ) value
              [   3,    7] (  45, DELI_CPAREN          ) )
              [   3,    8] (  48, DELI_OCURLY          ) {
3 -         Int last = 1;

              [   4,    1] (   1, RW_INT               ) Int
              [   4,    2] (  35, ID                   ) last
              [   4,    3] (  34, OPE_NEG              ) =
              [   4,    4] (  36, IDEN_INT             ) 1
              [   4,    5] (  51, DELI_SEMICOL         ) ;
4 -         Int penultimate = 1;

              [   5,    1] (   1, RW_INT               ) Int
              [   5,    2] (  35, ID                   ) penultimate
              [   5,    3] (  34, OPE_NEG              ) =
              [   5,    4] (  36, IDEN_INT             ) 1
              [   5,    5] (  51, DELI_SEMICOL         ) ;
5 -         Int next;

              [   6,    1] (   1, RW_INT               ) Int
              [   6,    2] (  35, ID                   ) next
              [   6,    3] (  51, DELI_SEMICOL         ) ;
6 -

7 -         If value == 1 {

              [   8,    1] (   8, RW_IF                ) If
              [   8,    2] (  35, ID                   ) value
              [   8,    3] (  33, OPE_REL              ) ==
              [   8,    4] (  36, IDEN_INT             ) 1
              [   8,    5] (  48, DELI_OCURLY          ) {
8 -             Return 1;

              [   9,    1] (  16, RW_RETURN            ) Return
              [   9,    2] (  36, IDEN_INT             ) 1
              [   9,     3] (  51, DELI_SEMICOL        ) ;
9 -         }

              [  10,     1] (  49, DELI_CCURLY         ) }
10 -         If value == 2 {

              [  11,     1] (   8, RW_IF               ) If
              [  11,     2] (  35, ID                  ) value
              [  11,     3] (  33, OPE_REL             ) ==
              [  11,     4] (  36, IDEN_INT            ) 2
              [  11,     5] (  48, DELI_OCURLY         ) {
11 -             Return 1;

              [  12,     1] (  16, RW_RETURN           ) Return
              [  12,     2] (  36, IDEN_INT            ) 1
              [  12,     3] (  51, DELI_SEMICOL        ) ;
12 -         }

              [  13,     1] (  49, DELI_CCURLY         ) }
13 -

14 -         Int count = 3;

              [  15,     1] (   1, RW_INT              ) Int
              [  15,     2] (  35, ID                  ) count
              [  15,     3] (  34, OPE_NEG             ) =
              [  15,     4] (  36, IDEN_INT            ) 3
              [  15,     5] (  51, DELI_SEMICOL        ) ;
15 -         While count <= value {

              [  16,     1] (  10, RW_WHILE            ) While
              [  16,     2] (  35, ID                  ) count
              [  16,     3] (  31, OPE_LE              ) <=
              [  16,     4] (  35, ID                  ) value
              [  16,     5] (  48, DELI_OCURLY         ) {
16 -             next = last + penultimate;

              [  17,     1] (  35, ID                  ) next
              [  17,     2] (  34, OPE_NEG             ) =
              [  17,     3] (  35, ID                  ) last
              [  17,     4] (  24, OPE_ADD             ) +
              [  17,     5] (  35, ID                  ) penultimate
              [  17,     6] (  51, DELI_SEMICOL        ) ;
17 -             penultimate = last;

              [  18,     1] (  35, ID                  ) penultimate
              [  18,     2] (  34, OPE_NEG             ) =
              [  18,     3] (  35, ID                  ) last
              [  18,     4] (  51, DELI_SEMICOL        ) ;
18 -             last = next;

              [  19,     1] (  35, ID                  ) last
              [  19,     2] (  34, OPE_NEG             ) =
              [  19,     3] (  35, ID                  ) next
              [  19,     4] (  51, DELI_SEMICOL        ) ;
19 -             count = count + 1;

              [  20,     1] (  35, ID                  ) count
```

```
                [  20,     2] (  34, OPE_NEG              ) =
                [  20,     3] (  35, ID                   ) count
                [  20,     4] (  24, OPE_ADD              ) +
                [  20,     5] (  36, IDEN_INT             ) 1
                [  20,     6] (  51, DELI_SEMICOL         ) ;
20 -           }

                [  21,     1] (  49, DELI_CCURLY          ) }
21 -

22 -           Return next;

                [  23,     1] (  16, RW_RETURN            ) Return
                [  23,     2] (  35, ID                   ) next
                [  23,     3] (  51, DELI_SEMICOL         ) ;
23 -       }

                [  24,     1] (  49, DELI_CCURLY          ) }
24 -

25 -       Int nthTerm;

                [  26,     1] (   1, RW_INT               ) Int
                [  26,     2] (  35, ID                   ) nthTerm
                [  26,     3] (  51, DELI_SEMICOL         ) ;
26 -       Int result;

                [  27,     1] (   1, RW_INT               ) Int
                [  27,     2] (  35, ID                   ) result
                [  27,     3] (  51, DELI_SEMICOL         ) ;
27 -

28 -       Get(nthTerm);

                [  29,     1] (  14, RW_GET               ) Get
                [  29,     2] (  44, DELI_OPAREN          ) (
                [  29,     3] (  35, ID                   ) nthTerm
                [  29,     4] (  45, DELI_CPAREN          ) )
                [  29,     5] (  51, DELI_SEMICOL         ) ;
29 -

30 -       result = fibonacci(value);

                [  31,     1] (  35, ID                   ) result
                [  31,     2] (  34, OPE_NEG              ) =
                [  31,     3] (  35, ID                   ) fibonacci
                [  31,     4] (  44, DELI_OPAREN          ) (
                [  31,     5] (  35, ID                   ) value
                [  31,     6] (  45, DELI_CPAREN          ) )
                [  31,     7] (  51, DELI_SEMICOL         ) ;
31 -       Show("%d", result);

                [  32,     1] (  15, RW_SHOW              ) Show
                [  32,     2] (  44, DELI_OPAREN          ) (
                [  32,     3] (  40, IDEN_STRING          ) "%d"
                [  32,     4] (  50, DELI_COMMA           ) ,
                [  32,     5] (  35, ID                   ) result
                [  32,     6] (  45, DELI_CPAREN          ) )
```

```
              [   32,     7] (   51, DELI_SEMICOL        ) ;
32 - }
              [   33,     1] (   49, DELI_CCURLY         ) }
              [   33,     1] (   57, OTHER_EOF           ) EOF
```

## 3 Shell Sort

### 3.1 Programa

```
Begin {
    Function Int shellSort(Array array[], Int n) {
        Int gap = n / 2;
        While gap > 0 {
            Int i = gap;
            From i To (n - 1) Increase 1 {
                Int temp = array[i];
                Int j = i;

                While j >= gap And array[j - gap] > temp {
                    array[j] = array[j - gap];
                    j = j - gap;
                }
                array[j] = temp;
            }
        }
        Return 0;
    }

    Array Int array[10];
    Int i = 0;

    From i To 9 Increase 1 {
        int input;
        get(input);
        array[i] = input;
    }

    shellSort(array, 10);

    From i To 9 Increase 1 {
        Show("%d ", array[i]);
    }
}
```

### 3.2 Resultado

```
1 - Begin {

              [    2,     1] (    6, RW_BEGIN            ) Begin
              [    2,     2] (   48, DELI_OCURLY         ) {
2 -     Function Int shellSort(Array array[], Int n) {

              [    3,     1] (   17, RW_FUNCTION         ) Function
              [    3,     2] (    1, RW_INT              ) Int
```

```
                  [    3,     3] (  35, ID                    ) shellSort
                  [    3,     4] (  44, DELI_OPAREN           ) (
                  [    3,     5] (  20, RW_ARRAY              ) Array
                  [    3,     6] (  35, ID                    ) array
                  [    3,     7] (  46, DELI_OBRAC            ) [
                  [    3,     8] (  47, DELI_CBRAC            ) ]
                  [    3,     9] (  50, DELI_COMMA            ) ,
                  [    3,    10] (   1, RW_INT                ) Int
                  [    3,    11] (  35, ID                    ) n
                  [    3,    12] (  45, DELI_CPAREN           ) )
                  [    3,    13] (  48, DELI_OCURLY           ) {
3 -             Int gap = n / 2;

                  [    4,     1] (   1, RW_INT                ) Int
                  [    4,     2] (  35, ID                    ) gap
                  [    4,     3] (  34, OPE_NEG               ) =
                  [    4,     4] (  35, ID                    ) n
                  [    4,     5] (  27, OPE_DIV               ) /
                  [    4,     6] (  36, IDEN_INT              ) 2
                  [    4,     7] (  51, DELI_SEMICOL          ) ;
4 -             While gap > 0 {

                  [    5,     1] (  10, RW_WHILE              ) While
                  [    5,     2] (  35, ID                    ) gap
                  [    5,     3] (  30, OPE_GT                ) >
                  [    5,     4] (  36, IDEN_INT              ) 0
                  [    5,     5] (  48, DELI_OCURLY           ) {
5 -                 Int i = gap;

                  [    6,     1] (   1, RW_INT                ) Int
                  [    6,     2] (  35, ID                    ) i
                  [    6,     3] (  34, OPE_NEG               ) =
                  [    6,     4] (  35, ID                    ) gap
                  [    6,     5] (  51, DELI_SEMICOL          ) ;
6 -                 From i To (n - 1) Increase 1 {

                  [    7,     1] (  11, RW_FROM               ) From
                  [    7,     2] (  35, ID                    ) i
                  [    7,     3] (  12, RW_TO                 ) To
                  [    7,     4] (  44, DELI_OPAREN           ) (
                  [    7,     5] (  35, ID                    ) n
                  [    7,     6] (  25, OPE_SUB               ) -
                  [    7,     7] (  36, IDEN_INT              ) 1
                  [    7,     8] (  45, DELI_CPAREN           ) )
                  [    7,     9] (  13, RW_INCREASE           ) Increase
                  [    7,    10] (  36, IDEN_INT              ) 1
                  [    7,    11] (  48, DELI_OCURLY           ) {
7 -                     Int temp = array[i];

                  [    8,     1] (   1, RW_INT                ) Int
                  [    8,     2] (  35, ID                    ) temp
                  [    8,     3] (  34, OPE_NEG               ) =
                  [    8,     4] (  35, ID                    ) array
                  [    8,     5] (  46, DELI_OBRAC            ) [
                  [    8,     6] (  35, ID                    ) i
                  [    8,     7] (  47, DELI_CBRAC            ) ]
                  [    8,     8] (  51, DELI_SEMICOL          ) ;
8 -                     Int j = i;
```

```
                [    9,     1] (    1, RW_INT               ) Int
                [    9,     2] (   35, ID                   ) j
                [    9,     3] (   34, OPE_NEG              ) =
                [    9,     4] (   35, ID                   ) i
                [    9,     5] (   51, DELI_SEMICOL         ) ;
9 -

10 -                   While j >= gap And array[j - gap] > temp {

                [   11,     1] (   10, RW_WHILE             ) While
                [   11,     2] (   35, ID                   ) j
                [   11,     3] (   32, OPE_GE               ) >=
                [   11,     4] (   35, ID                   ) gap
                [   11,     5] (   22, OPE_CONJ             ) And
                [   11,     6] (   35, ID                   ) array
                [   11,     7] (   46, DELI_OBRAC           ) [
                [   11,     8] (   35, ID                   ) j
                [   11,     9] (   25, OPE_SUB              ) -
                [   11,    10] (   35, ID                   ) gap
                [   11,    11] (   47, DELI_CBRAC           ) ]
                [   11,    12] (   30, OPE_GT               ) >
                [   11,    13] (   35, ID                   ) temp
                [   11,    14] (   48, DELI_OCURLY          ) {
11 -                       array[j] = array[j - gap];

                [   12,     1] (   35, ID                   ) array
                [   12,     2] (   46, DELI_OBRAC           ) [
                [   12,     3] (   35, ID                   ) j
                [   12,     4] (   47, DELI_CBRAC           ) ]
                [   12,     5] (   34, OPE_NEG              ) =
                [   12,     6] (   35, ID                   ) array
                [   12,     7] (   46, DELI_OBRAC           ) [
                [   12,     8] (   35, ID                   ) j
                [   12,     9] (   25, OPE_SUB              ) -
                [   12,    10] (   35, ID                   ) gap
                [   12,    11] (   47, DELI_CBRAC           ) ]
                [   12,    12] (   51, DELI_SEMICOL         ) ;
12 -                       j = j - gap;

                [   13,     1] (   35, ID                   ) j
                [   13,     2] (   34, OPE_NEG              ) =
                [   13,     3] (   35, ID                   ) j
                [   13,     4] (   25, OPE_SUB              ) -
                [   13,     5] (   35, ID                   ) gap
                [   13,     6] (   51, DELI_SEMICOL         ) ;
13 -                   }

                [   14,     1] (   49, DELI_CCURLY          ) }
14 -                   array[j] = temp;

                [   15,     1] (   35, ID                   ) array
                [   15,     2] (   46, DELI_OBRAC           ) [
                [   15,     3] (   35, ID                   ) j
                [   15,     4] (   47, DELI_CBRAC           ) ]
                [   15,     5] (   34, OPE_NEG              ) =
                [   15,     6] (   35, ID                   ) temp
                [   15,     7] (   51, DELI_SEMICOL         ) ;
```

```
15 -             }

           [  16,     1] (  49, DELI_CCURLY          ) }
16 -         }

           [  17,     1] (  49, DELI_CCURLY          ) }
17 -         Return 0;

           [  18,     1] (  16, RW_RETURN            ) Return
           [  18,     2] (  36, IDEN_INT             ) 0
           [  18,     3] (  51, DELI_SEMICOL         ) ;
18 -     }

           [  19,     1] (  49, DELI_CCURLY          ) }
19 -

20 -     Array Int array[10];

           [  21,     1] (  20, RW_ARRAY             ) Array
           [  21,     2] (   1, RW_INT               ) Int
           [  21,     3] (  35, ID                   ) array
           [  21,     4] (  46, DELI_OBRAC           ) [
           [  21,     5] (  36, IDEN_INT             ) 10
           [  21,     6] (  47, DELI_CBRAC           ) ]
           [  21,     7] (  51, DELI_SEMICOL         ) ;
21 -     Int i = 0;

           [  22,     1] (   1, RW_INT               ) Int
           [  22,     2] (  35, ID                   ) i
           [  22,     3] (  34, OPE_NEG              ) =
           [  22,     4] (  36, IDEN_INT             ) 0
           [  22,     5] (  51, DELI_SEMICOL         ) ;
22 -

23 -     From i To 9 Increase 1 {

           [  24,     1] (  11, RW_FROM              ) From
           [  24,     2] (  35, ID                   ) i
           [  24,     3] (  12, RW_TO                ) To
           [  24,     4] (  36, IDEN_INT             ) 9
           [  24,     5] (  13, RW_INCREASE          ) Increase
           [  24,     6] (  36, IDEN_INT             ) 1
           [  24,     7] (  48, DELI_OCURLY          ) {
24 -         int input;

           [  25,     1] (  35, ID                   ) int
           [  25,     2] (  35, ID                   ) input
           [  25,     3] (  51, DELI_SEMICOL         ) ;
25 -         get(input);

           [  26,     1] (  35, ID                   ) get
           [  26,     2] (  44, DELI_OPAREN          ) (
           [  26,     3] (  35, ID                   ) input
           [  26,     4] (  45, DELI_CPAREN          ) )
           [  26,     5] (  51, DELI_SEMICOL         ) ;
26 -         array[i] = input;

           [  27,     1] (  35, ID                   ) array
```

```
            [  27,     2] (  46, DELI_OBRAC          ) [
            [  27,     3] (  35, ID                  ) i
            [  27,     4] (  47, DELI_CBRAC          ) ]
            [  27,     5] (  34, OPE_NEG             ) =
            [  27,     6] (  35, ID                  ) input
            [  27,     7] (  51, DELI_SEMICOL        ) ;
27 -     }

            [  28,     1] (  49, DELI_CCURLY         ) }
28 -

29 -     shellSort(array, 10);

            [  30,     1] (  35, ID                  ) shellSort
            [  30,     2] (  44, DELI_OPAREN         ) (
            [  30,     3] (  35, ID                  ) array
            [  30,     4] (  50, DELI_COMMA          ) ,
            [  30,     5] (  36, IDEN_INT            ) 10
            [  30,     6] (  45, DELI_CPAREN         ) )
            [  30,     7] (  51, DELI_SEMICOL        ) ;
30 -

31 -     From i To 9 Increase 1 {

            [  32,     1] (  11, RW_FROM             ) From
            [  32,     2] (  35, ID                  ) i
            [  32,     3] (  12, RW_TO               ) To
            [  32,     4] (  36, IDEN_INT            ) 9
            [  32,     5] (  13, RW_INCREASE         ) Increase
            [  32,     6] (  36, IDEN_INT            ) 1
            [  32,     7] (  48, DELI_OCURLY         ) {
32 -         Show("%d ", array[i]);

            [  33,     1] (  15, RW_SHOW             ) Show
            [  33,     2] (  44, DELI_OPAREN         ) (
            [  33,     3] (  40, IDEN_STRING         ) "%d "
            [  33,     4] (  50, DELI_COMMA          ) ,
            [  33,     5] (  35, ID                  ) array
            [  33,     6] (  46, DELI_OBRAC          ) [
            [  33,     7] (  35, ID                  ) i
            [  33,     8] (  47, DELI_CBRAC          ) ]
            [  33,     9] (  45, DELI_CPAREN         ) )
            [  33,    10] (  51, DELI_SEMICOL        ) ;
33 -     }

            [  34,     1] (  49, DELI_CCURLY         ) }
34 - }
            [  35,     1] (  49, DELI_CCURLY         ) }
            [  35,     1] (  57, OTHER_EOF           ) EOF
```