

# Relatório de CES-41 – Compiladores

## Laboratório 6

Juan de Castro Pessoa  
Thiago Ribeiro Ramos  
COMP -14  
Professor Mokarzel  
16 de Dezembro de 2013

### ASSUNTO

Nesse laboratório foi elaborado um interpretador para a linguagem COMP-ITA-2013, com o auxílio da ferramenta Yacc e do código produzido no laboratório via Flex, a partir do gerador de código intermediário feito no laboratório anterior.

### RESULTADOS

#### Programa para determinar o Fibonnaci(n)

```
programa teste;
var
    int a;

funcao int fibonnaci(int x);
var int p1, p2, cont, temp;
{
    p1 := 0;
    p2 := 1;
    para(cont:=0;cont<x;cont:=cont+1){
        temp := p2;
        p2 := p1+p2;
        p1 := temp;
    }
    retornar p1;
}

{
    ler(a);
    escrever("O ",a," termo de Fibonnaci eh ", fibonnaci(a));
}
```

Para diferentes entradas:

15

INTERPRETADOR:

O 15 termo de Fibonnaci eh 610

0

INTERPRETADOR:

O 0 termo de Fibonnaci eh 0

4

INTERPRETADOR:

O 4 termo de Fibonnaci eh 3

40

INTERPRETADOR:

O 40 termo de Fibonnaci eh 102334155

Vale lembrar que como só cabem valores int, tem um limite para esse resultado.

### Programa com Bubble Sort

```
programa teste;

var
    int cx, cy;
    int tos[20];

procedimento swap(int a1, int a2);
var int temp;
{
    temp := tos[a1];
    tos[a1] := tos[a2];
    tos[a2] := temp;
    retornar;
}

procedimento bubblesort();
var
    int cx, cy;
{
    para(cx:=0;cx<20;cx:=cx+1){
        para(cy:=0;cy<19-cx;cy:=cy+1){
            se(tos[cy] > tos[cy+1]){
                chamar swap(cy,cy+1);
            }
        }
    }
    retornar;
}

procedimento imprimirtabela();
var int temp;
{
    para(temp:=0;temp<19;temp:=temp+1){
        escrever(tos[temp]," ");
    }
    escrever(tos[19]);
    retornar;
}

{
    cy:=1;
```

```

    para(cx:=0;cx<20;cx:=cx+1){
        tos[cx] := cy;
        cy := (cy*17)%23;
    }
    chamar imprimirtabela();
    chamar bubblesort();
    escrever("Depois de ordenado:");
    chamar imprimirtabela();
}

```

INTERPRETADOR:

```

1
17
13
14
8
21
12
20
18
7
4
22
6
10
9
15
2
11
3
5
Depois de ordenado:
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
17
18
20
21
22

```

### **Programa encontrador de quadrados perfeitos (o primeiro da lista)**

```

programa encontrarmaioria;

var
    logic s;
    int resultado;
    int tos[20];

```

```

funcao int raiz(int l);
var int temp;
{
    temp :=0;
    enquanto(temp*temp<=l){
        se(temp*temp=l){
            resultado := l;
            retornar temp;
        }
        temp := temp+1;
    }
    retornar ~1;
}

funcao logic temquadrado perfeito();
var int t;
{
    para(t:=0;t<20;t:=t+1){
        se(raiz(tos[t])!=~1){
            retornar verdade;
        }
    }
    retornar falso;
}

procedimento carregarvetor();
var int t;
{
    t:=0;
    repetir{
        ler(tos[t]);
        t:=t+1;
    }enquanto(t<20);
    retornar;
}

{
    chamar carregarvetor();
    s := temquadrado perfeito();
    se(s){
        escrever("Tem quadrado perfeito: ",resultado);
    }senao{
        escrever("Nao tem quadrado perfeito");
    }
}

```

Para diferentes entradas:

51 23 39 80 50 65 73 82 490 14
--------------------------------

INTERPRETADOR:
----------------

Nao tem quadrado perfeito
---------------------------

51 23 39 80 50 64 73 82 490 14
--------------------------------

INTERPRETADOR:
----------------

Tem quadrado perfeito: 64
---------------------------

1 5 90 81 49 64 73 121 490 14
-------------------------------

INTERPRETADOR:
----------------

Tem quadrado perfeito: 1
--------------------------

### Determinador de raízes quadradas de números reais

```
programa encontrarmaioria;

var
    int qtde, i;
    real val;

funcao real raiz(real l, real eps);

var real temp;
{
    se(l<0){
        retornar 0;
    }
    temp := 1;
    enquanto(temp*temp<=l-eps||temp*temp>=l+eps){
        temp := (temp+l/temp)/2.0;
    }
    retornar temp;
}

{
    ler(qtde);
    para(i:=0;i<qtde;i:=i+1){
        ler(val);
        escrever("Raiz de ",val," eh aprox ",raiz(val,0.1));
    }
}
```

Para dferentes entradas (para valores negativos, que nao tem raiz, ele imprime 0):

7 1 2 20 7 0.4 150 121
------------------------

INTERPRETADOR:
----------------

Raiz de 1 eh aprox 1
----------------------

Raiz de 2 eh aprox 1.41667
----------------------------

Raiz de 20 eh aprox 4.47831
-----------------------------

Raiz de 7 eh aprox 2.65489
----------------------------

Raiz de 0.4 eh aprox 0.7
--------------------------

Raiz de 150 eh aprox 12.2481
------------------------------

Raiz de 121 eh aprox 11.0002
------------------------------

3 1000000 0.01 1048576
------------------------

INTERPRETADOR:
----------------

Raiz de 1e+06 eh aprox 1000
-----------------------------

Raiz de 0.01 eh aprox 0.262401
--------------------------------

Raiz de 1.04858e+06 eh aprox 1024
-----------------------------------

4 1.21 13 0 -5
INTERPRETADOR: Raiz de 1.21 eh aprox 1.105 Raiz de 13 eh aprox 3.60635 Raiz de 0 eh aprox 0.25 Raiz de -5 eh aprox 0