

# Relatório de CES-41 – Compiladores

## Laboratório 5

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

### ASSUNTO

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

### RESULTADOS

#### Programa mais simples

```
programa teste;

var
    int a, b, c, d;

{
    a := b+c;
    c := c/d;
    d := a*c;
    d := 5*a;
    b := 15-d;
    a := a*3+c%d-4/a;
}
```

Quadruplas do modulo teste:

- 1) OPENMOD, (MODULO, teste), (IDLE), (IDLE)
- 2) MAIS, (VAR, b), (VAR, c), (VAR, ##1)
- 3) ATRIB, (VAR, ##1), (IDLE), (VAR, a)
- 4) DIV, (VAR, c), (VAR, d), (VAR, ##2)
- 5) ATRIB, (VAR, ##2), (IDLE), (VAR, c)
- 6) MULT, (VAR, a), (VAR, c), (VAR, ##3)
- 7) ATRIB, (VAR, ##3), (IDLE), (VAR, d)
- 8) RESTO, (INT, 5), (VAR, a), (VAR, ##4)
- 9) ATRIB, (VAR, ##4), (IDLE), (VAR, d)
- 10) MENOS, (INT, 15), (VAR, d), (VAR, ##5)
- 11) ATRIB, (VAR, ##5), (IDLE), (VAR, b)
- 12) MULT, (VAR, a), (INT, 3), (VAR, ##6)
- 13) RESTO, (VAR, c), (VAR, d), (VAR, ##7)
- 14) MAIS, (VAR, ##6), (VAR, ##7), (VAR, ##8)
- 15) DIV, (INT, 4), (VAR, a), (VAR, ##9)
- 16) MENOS, (VAR, ##8), (VAR, ##9), (VAR, ##10)
- 17) ATRIB, (VAR, ##10), (IDLE), (VAR, a)
- 18) EXIT, (IDLE), (IDLE), (IDLE)

#### Utilizando Se

```
programa teste;
```

```

var
    int a;
    logic g, h;
    carac i;

{
    a := 5;
    g :=falso;
    h :=verdade;
    i := 0;
    se(a<8 && g!=h){
        i := i+1;
    }
}

```

Quadruplas do modulo teste:

- 1) OPENMOD, (MODULO, teste), (IDLE), (IDLE)
- 2) ATRIB, (INT, 5), (IDLE), (VAR, a)
- 3) ATRIB, (LOGIC, 0), (IDLE), (VAR, g)
- 4) ATRIB, (LOGIC, 1), (IDLE), (VAR, h)
- 5) ATRIB, (INT, 0), (IDLE), (VAR, i)
- 6) LT, (VAR, a), (INT, 8), (VAR, ##1)
- 7) NE, (VAR, g), (VAR, h), (VAR, ##2)
- 8) AND, (VAR, ##1), (VAR, ##2), (VAR, ##3)
- 9) JF, (VAR, ##3), (IDLE), (ROTULO, 12)
- 10) MAIS, (VAR, i), (INT, 1), (VAR, ##4)
- 11) ATRIB, (VAR, ##4), (IDLE), (VAR, i)
- 12) NOP, (IDLE), (IDLE), (IDLE)
- 13) EXIT, (IDLE), (IDLE), (IDLE)

Como esperado, se a condição não foi satisfeita, há um salto para um NOP depois do fim do fluxo.

### Utilizando Para

```

programa teste;

var
    int c, d;
    logic e;
    real m;

{
    d := 3;
    e := falso;
    para(c:=0;c<d;c:=c+1){
        d := d-1;
        m := m*2;
    }
}

```

Quadruplas do modulo teste:

- 1) OPENMOD, (MODULO, teste), (IDLE), (IDLE)
- 2) ATRIB, (INT, 3), (IDLE), (VAR, d)
- 3) ATRIB, (LOGIC, 0), (IDLE), (VAR, e)
- 4) ATRIB, (INT, 0), (IDLE), (VAR, c)
- 5) NOP, (IDLE), (IDLE), (IDLE)

```

6) LT, (VAR, c), (VAR, d), (VAR, ##1)
7) JF, (VAR, ##1), (IDLE), (ROTULO, 15)
8) MENOS, (VAR, d), (INT, 1), (VAR, ##3)
9) ATRIB, (VAR, ##3), (IDLE), (VAR, d)
10) MULT, (VAR, m), (INT, 2), (VAR, ##4)
11) ATRIB, (VAR, ##4), (IDLE), (VAR, m)
12) MAIS, (VAR, c), (INT, 1), (VAR, ##2)
13) ATRIB, (VAR, ##2), (IDLE), (VAR, c)
14) JUMP, (IDLE), (IDLE), (ROTULO, 5)
15) NOP, (IDLE), (IDLE), (IDLE)
16) EXIT, (IDLE), (IDLE), (IDLE)

```

Como esperado, o código faz a atribuição uma vez, depois sempre faz a verificação se a condição é verdadeira. Se não, faz um JUMP pro NOP de código 15, para terminar o loop. Do contrário, executa todos os comandos e ao final a atribuição do PARA, e então volta ao ponto de conferência da condição da expressão.

### Misturando vários gerenciadores de fluxo

```

programa teste;

var
    int a, b, c, d;
    logic e, f, g, h;
    char i;
    real m;

{
    a := 5;
    b := 0;
    c := b+a;
    d := a*41;
    e := falso;
    f := falso;
    g := falso;
    h := verdade;
    m := 0.3;
    i := 'a';
    se(a<8 && g!=h){
        i := i+1;
        enquanto(b!=10 || f){
            b:=b+1;
        }
    }
    para(c:=0;c>=d;c:=c+1){
        repetir{
            escrever(c,d);
            d := d-1;
            m := m*2;
        }enquanto(d>0 && e);
    }
}

```

Quadruplas do modulo teste:

```

1) OPENMOD, (MODULO, teste), (IDLE), (IDLE)
2) ATRIB, (INT, 5), (IDLE), (VAR, a)
3) ATRIB, (INT, 0), (IDLE), (VAR, b)
4) MAIS, (VAR, b), (VAR, a), (VAR, ##1)

```

```

5) ATRIB, (VAR, ##1), (IDLE), (VAR, c)
6) MULT, (VAR, a), (INT, 41), (VAR, ##2)
7) ATRIB, (VAR, ##2), (IDLE), (VAR, d)
8) ATRIB, (LOGIC, 0), (IDLE), (VAR, e)
9) ATRIB, (LOGIC, 0), (IDLE), (VAR, f)
10) ATRIB, (LOGIC, 0), (IDLE), (VAR, g)
11) ATRIB, (LOGIC, 1), (IDLE), (VAR, h)
12) ATRIB, (REAL, 0.3), (IDLE), (VAR, m)
13) ATRIB, (CARAC, a), (IDLE), (VAR, i)
14) LT, (VAR, a), (INT, 8), (VAR, ##3)
15) NE, (VAR, g), (VAR, h), (VAR, ##4)
16) AND, (VAR, ##3), (VAR, ##4), (VAR, ##5)
17) JF, (VAR, ##5), (IDLE), (ROTULO, 28)
18) MAIS, (VAR, i), (INT, 1), (VAR, ##6)
19) ATRIB, (VAR, ##6), (IDLE), (VAR, i)
20) NOP, (IDLE), (IDLE), (IDLE)
21) NE, (VAR, b), (INT, 10), (VAR, ##7)
22) OR, (VAR, ##7), (VAR, f), (VAR, ##8)
23) JF, (VAR, ##8), (IDLE), (ROTULO, 27)
24) MAIS, (VAR, b), (INT, 1), (VAR, ##9)
25) ATRIB, (VAR, ##9), (IDLE), (VAR, b)
26) JUMP, (IDLE), (IDLE), (ROTULO, 20)
27) NOP, (IDLE), (IDLE), (IDLE)
28) NOP, (IDLE), (IDLE), (IDLE)
29) ATRIB, (INT, 0), (IDLE), (VAR, c)
30) NOP, (IDLE), (IDLE), (IDLE)
31) GE, (VAR, c), (VAR, d), (VAR, ##10)
32) JF, (VAR, ##10), (IDLE), (ROTULO, 49)
33) NOP, (IDLE), (IDLE), (IDLE)
34) PARAM, (VAR, c), (IDLE), (IDLE)
35) PARAM, (VAR, d), (IDLE), (IDLE)
36) WRITE, (INT, 2), (IDLE), (IDLE)
37) MENOS, (VAR, d), (INT, 1), (VAR, ##12)
38) ATRIB, (VAR, ##12), (IDLE), (VAR, d)
39) MULT, (VAR, m), (INT, 2), (VAR, ##13)
40) ATRIB, (VAR, ##13), (IDLE), (VAR, m)
41) GT, (VAR, d), (INT, 0), (VAR, ##14)
42) AND, (VAR, ##14), (VAR, e), (VAR, ##15)
43) JF, (VAR, ##15), (IDLE), (ROTULO, 45)
44) JUMP, (IDLE), (IDLE), (ROTULO, 33)
45) NOP, (IDLE), (IDLE), (IDLE)
46) MAIS, (VAR, c), (INT, 1), (VAR, ##11)
47) ATRIB, (VAR, ##11), (IDLE), (VAR, c)
48) JUMP, (IDLE), (IDLE), (ROTULO, 30)
49) NOP, (IDLE), (IDLE), (IDLE)
50) EXIT, (IDLE), (IDLE), (IDLE)

```

## Utilizando ponteiros

```

programa teste;

var
    real a, b[3,4], c[5,6,7];

{
    c[0,0,0] :=0;
    b[1,2] := a;
    a := c[4,3,2];
}

```

Quadruplas do modulo teste:

```
1) OPENMOD, (MODULO, teste), (IDLE), (IDLE)
2) IND, (INT, 0), (IDLE), (IDLE)
3) IND, (INT, 0), (IDLE), (IDLE)
4) IND, (INT, 0), (IDLE), (IDLE)
5) INDEX, (VAR, c), (INT, 3), (VAR, ##1)
6) ATRIBPONT, (INT, 0), (IDLE), (VAR, ##1)
7) IND, (INT, 1), (IDLE), (IDLE)
8) IND, (INT, 2), (IDLE), (IDLE)
9) INDEX, (VAR, b), (INT, 2), (VAR, ##2)
10) ATRIBPONT, (VAR, a), (IDLE), (VAR, ##2)
11) IND, (INT, 4), (IDLE), (IDLE)
12) IND, (INT, 3), (IDLE), (IDLE)
13) IND, (INT, 2), (IDLE), (IDLE)
14) INDEX, (VAR, c), (INT, 3), (VAR, ##3)
15) CONTAPONT, (VAR, ##3), (IDLE), (VAR, ##4)
16) ATRIB, (VAR, ##4), (IDLE), (VAR, a)
17) EXIT, (IDLE), (IDLE), (IDLE)
```

Como esperado, o CONTAPONT passa o conteúdo da variável indexada para uma variável comum, para que ela possa ser utilizada pelos outros comandos sem problemas. O Atribpoint cuida de fazer a atribuição considerando qual ponto da variável o valor deve ser alocado. O IND adiciona valores a uma suposta pilha para o INDEX ler depois.

### Utilizando funções e procedimentos

```
programa teste;

var
    int b;
    real e;

funcao real func(int x, carac c);
{
    x := x*x+c;
    retornar x+0.5;
}

procedimento proc(real r);
{
    b := b*4;
    se(r>0.5){
        b:=b+1;
    }senao{
        b:=b-1;
    }
    retornar;
}

{
    e:=4;
    e := func(4, 'y');
    chamar proc(e);
}
```

Quadruplas do modulo teste:

```

1) OPENMOD, (MODULO, teste), (IDLE), (IDLE)
14) ATRIB, (INT, 4), (IDLE), (VAR, e)
15) PARAM, (INT, 4), (IDLE), (IDLE)
16) PARAM, (CARAC, y), (IDLE), (IDLE)
17) CALL, (FUNCAO, func), (INT, 2), (VAR, ##8)
18) ATRIB, (VAR, ##8), (IDLE), (VAR, e)
19) PARAM, (VAR, e), (IDLE), (IDLE)
20) CALL, (FUNCAO, proc), (INT, 1), (IDLE)
21) EXIT, (IDLE), (IDLE), (IDLE)

```

Quadruplas do modulo func:

```

1) OPENMOD, (MODULO, func), (IDLE), (IDLE)
2) MULT, (VAR, x), (VAR, x), (VAR, ##1)
3) MAIS, (VAR, ##1), (VAR, c), (VAR, ##2)
4) ATRIB, (VAR, ##2), (IDLE), (VAR, x)
5) MAIS, (VAR, x), (REAL, 0.5), (VAR, ##3)
6) RETORNO, (VAR, ##3), (IDLE), (IDLE)

```

Quadruplas do modulo proc:

```

1) OPENMOD, (MODULO, proc), (IDLE), (IDLE)
2) MULT, (VAR, b), (INT, 4), (VAR, ##4)
3) ATRIB, (VAR, ##4), (IDLE), (VAR, b)
4) GT, (VAR, r), (REAL, 0.5), (VAR, ##5)
5) JF, (VAR, ##5), (IDLE), (ROTULO, 9)
6) MAIS, (VAR, b), (INT, 1), (VAR, ##6)
7) ATRIB, (VAR, ##6), (IDLE), (VAR, b)
8) JUMP, (IDLE), (IDLE), (ROTULO, 12)
9) NOP, (IDLE), (IDLE), (IDLE)
10) MENOS, (VAR, b), (INT, 1), (VAR, ##7)
11) ATRIB, (VAR, ##7), (IDLE), (VAR, b)
12) NOP, (IDLE), (IDLE), (IDLE)
13) RETORNO, (IDLE), (IDLE), (IDLE)

```

## Em um código mais complexo

```

programa teste;

var
    int a, resultado,cx,cy;
    int h[5,6], tos[20];

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;
}

```

```

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;
}

procedimento preenchertabelah(int inicial, int razaox, int razaoy);

var int cx, cy;
{
    para(cx:=0;cx<5;cx:=cx+1){
        para(cy:=0;cy<6;cy:=cy+1){
            h[cx,cy] := inicial+cx*razaox+cy*razaoy;
        }
    }
    retornar;
}

{
    ler(a);
    escrever("0 ",a," termo de Fibonnaci eh", fibonnaci(a));
    cy:=1;
    para(cx:=0;cx<20;cx:=cx+1){
        tos[cx] := cy;
        cy := (cy*17)%23;
    }
    chamar imprimirtabela();
    chamar bubblesort();
    chamar imprimirtabela();
    resultado := fibonnaci(10);
    se(h[1,3]>~5){
        escrever("Resultado eh ",resultado);
    }
}

```

}  
}

#### Quadruplas do modulo teste:

- 1) OPENMOD, (MODULO, teste), (IDLE), (IDLE)
- 2) PARAM, (VAR, a), (IDLE), (IDLE)
- 3) READ, (INT, 1), (IDLE), (IDLE)
- 4) PARAM, (CADEIA, "O "), (IDLE), (IDLE)
- 5) PARAM, (VAR, a), (IDLE), (IDLE)
- 6) PARAM, (CADEIA, " termo de Fibonnaci eh"), (IDLE), (IDLE)
- 7) PARAM, (VAR, a), (IDLE), (IDLE)
- 8) CALL, (FUNCAO, fibonnaci), (INT, 1), (VAR, ##37)
- 9) PARAM, (VAR, ##37), (IDLE), (IDLE)
- 10) WRITE, (INT, 4), (IDLE), (IDLE)
- 11) ATRIB, (INT, 1), (IDLE), (VAR, cy)
- 12) ATRIB, (INT, 0), (IDLE), (VAR, cx)
- 13) NOP, (IDLE), (IDLE), (IDLE)
- 14) LT, (VAR, cx), (INT, 20), (VAR, ##38)
- 15) JF, (VAR, ##38), (IDLE), (ROTULO, 25)
- 16) IND, (VAR, cx), (IDLE), (IDLE)
- 17) INDEX, (VAR, tos), (INT, 1), (VAR, ##40)
- 18) ATRIBPONT, (VAR, cy), (IDLE), (VAR, ##40)
- 19) MULT, (VAR, cy), (INT, 17), (VAR, ##41)
- 20) RESTO, (VAR, ##41), (INT, 23), (VAR, ##42)
- 21) ATRIB, (VAR, ##42), (IDLE), (VAR, cy)
- 22) MAIS, (VAR, cx), (INT, 1), (VAR, ##39)
- 23) ATRIB, (VAR, ##39), (IDLE), (VAR, cx)
- 24) JUMP, (IDLE), (IDLE), (ROTULO, 13)
- 25) NOP, (IDLE), (IDLE), (IDLE)
- 26) CALL, (FUNCAO, imprimirtabela), (INT, 0), (IDLE)
- 27) CALL, (FUNCAO, bubblesort), (INT, 0), (IDLE)
- 28) CALL, (FUNCAO, imprimirtabela), (INT, 0), (IDLE)
- 29) PARAM, (INT, 10), (IDLE), (IDLE)
- 30) CALL, (FUNCAO, fibonnaci), (INT, 1), (VAR, ##43)
- 31) ATRIB, (VAR, ##43), (IDLE), (VAR, resultado)
- 32) IND, (INT, 1), (IDLE), (IDLE)
- 33) IND, (INT, 3), (IDLE), (IDLE)
- 34) INDEX, (VAR, h), (INT, 2), (VAR, ##44)
- 35) CONTAPONT, (VAR, ##44), (IDLE), (VAR, ##45)
- 36) MENUN, (INT, 5), (IDLE), (VAR, ##46)
- 37) GT, (VAR, ##45), (VAR, ##46), (VAR, ##47)
- 38) JF, (VAR, ##47), (IDLE), (ROTULO, 42)
- 39) PARAM, (CADEIA, "Resultado eh "), (IDLE), (IDLE)
- 40) PARAM, (VAR, resultado), (IDLE), (IDLE)
- 41) WRITE, (INT, 2), (IDLE), (IDLE)
- 42) NOP, (IDLE), (IDLE), (IDLE)
- 43) EXIT, (IDLE), (IDLE), (IDLE)

#### Quadruplas do modulo fibonnaci:

- 1) OPENMOD, (MODULO, fibonnaci), (IDLE), (IDLE)
- 2) ATRIB, (INT, 0), (IDLE), (VAR, p1)
- 3) ATRIB, (INT, 1), (IDLE), (VAR, p2)
- 4) ATRIB, (INT, 0), (IDLE), (VAR, cont)
- 5) NOP, (IDLE), (IDLE), (IDLE)
- 6) LT, (VAR, cont), (VAR, x), (VAR, ##1)
- 7) JF, (VAR, ##1), (IDLE), (ROTULO, 15)
- 8) ATRIB, (VAR, p2), (IDLE), (VAR, temp)
- 9) MAIS, (VAR, p1), (VAR, p2), (VAR, ##3)
- 10) ATRIB, (VAR, ##3), (IDLE), (VAR, p2)
- 11) ATRIB, (VAR, temp), (IDLE), (VAR, p1)



- 12) MAIS, (VAR, cont), (INT, 1), (VAR, ##2)
- 13) ATRIB, (VAR, ##2), (IDLE), (VAR, cont)
- 14) JUMP, (IDLE), (IDLE), (ROTULO, 5)
- 15) NOP, (IDLE), (IDLE), (IDLE)
- 16) RETORNO, (VAR, p1), (IDLE), (IDLE)

Quadruplas do modulo swap:

- 1) OPENMOD, (MODULO, swap), (IDLE), (IDLE)
- 2) IND, (VAR, a1), (IDLE), (IDLE)
- 3) INDEX, (VAR, tos), (INT, 1), (VAR, ##4)
- 4) CONTAPONT, (VAR, ##4), (IDLE), (VAR, ##5)
- 5) ATRIB, (VAR, ##5), (IDLE), (VAR, temp)
- 6) IND, (VAR, a1), (IDLE), (IDLE)
- 7) INDEX, (VAR, tos), (INT, 1), (VAR, ##6)
- 8) IND, (VAR, a2), (IDLE), (IDLE)
- 9) INDEX, (VAR, tos), (INT, 1), (VAR, ##7)
- 10) CONTAPONT, (VAR, ##7), (IDLE), (VAR, ##8)
- 11) ATRIBPONT, (VAR, ##8), (IDLE), (VAR, ##6)
- 12) IND, (VAR, a2), (IDLE), (IDLE)
- 13) INDEX, (VAR, tos), (INT, 1), (VAR, ##9)
- 14) ATRIBPONT, (VAR, temp), (IDLE), (VAR, ##9)
- 15) RETORNO, (IDLE), (IDLE), (IDLE)

Quadruplas do modulo bubblesort:

- 1) OPENMOD, (MODULO, bubblesort), (IDLE), (IDLE)
- 2) ATRIB, (INT, 0), (IDLE), (VAR, cx)
- 3) NOP, (IDLE), (IDLE), (IDLE)
- 4) LT, (VAR, cx), (INT, 20), (VAR, ##10)
- 5) JF, (VAR, ##10), (IDLE), (ROTULO, 32)
- 6) ATRIB, (INT, 0), (IDLE), (VAR, cy)
- 7) NOP, (IDLE), (IDLE), (IDLE)
- 8) MENOS, (INT, 19), (VAR, cx), (VAR, ##12)
- 9) LT, (VAR, cy), (VAR, ##12), (VAR, ##13)
- 10) JF, (VAR, ##13), (IDLE), (ROTULO, 28)
- 11) IND, (VAR, cy), (IDLE), (IDLE)
- 12) INDEX, (VAR, tos), (INT, 1), (VAR, ##15)
- 13) CONTAPONT, (VAR, ##15), (IDLE), (VAR, ##16)
- 14) MAIS, (VAR, cy), (INT, 1), (VAR, ##17)
- 15) IND, (VAR, ##17), (IDLE), (IDLE)
- 16) INDEX, (VAR, tos), (INT, 1), (VAR, ##18)
- 17) CONTAPONT, (VAR, ##18), (IDLE), (VAR, ##19)
- 18) GT, (VAR, ##16), (VAR, ##19), (VAR, ##20)
- 19) JF, (VAR, ##20), (IDLE), (ROTULO, 24)
- 20) PARAM, (VAR, cy), (IDLE), (IDLE)
- 21) MAIS, (VAR, cy), (INT, 1), (VAR, ##21)
- 22) PARAM, (VAR, ##21), (IDLE), (IDLE)
- 23) CALL, (FUNCAO, swap), (INT, 2), (IDLE)
- 24) NOP, (IDLE), (IDLE), (IDLE)
- 25) MAIS, (VAR, cy), (INT, 1), (VAR, ##14)
- 26) ATRIB, (VAR, ##14), (IDLE), (VAR, cy)
- 27) JUMP, (IDLE), (IDLE), (ROTULO, 7)
- 28) NOP, (IDLE), (IDLE), (IDLE)
- 29) MAIS, (VAR, cx), (INT, 1), (VAR, ##11)
- 30) ATRIB, (VAR, ##11), (IDLE), (VAR, cx)
- 31) JUMP, (IDLE), (IDLE), (ROTULO, 3)
- 32) NOP, (IDLE), (IDLE), (IDLE)
- 33) RETORNO, (IDLE), (IDLE), (IDLE)

Quadruplas do modulo imprimirtabela:

- 1) OPENMOD, (MODULO, imprimirtabela), (IDLE), (IDLE)
- 2) ATRIB, (INT, 0), (IDLE), (VAR, temp)
- 3) NOP, (IDLE), (IDLE), (IDLE)
- 4) LT, (VAR, temp), (INT, 19), (VAR, ##22)
- 5) JF, (VAR, ##22), (IDLE), (ROTULO, 15)
- 6) IND, (VAR, temp), (IDLE), (IDLE)
- 7) INDEX, (VAR, tos), (INT, 1), (VAR, ##24)
- 8) CONTAPONT, (VAR, ##24), (IDLE), (VAR, ##25)
- 9) PARAM, (VAR, ##25), (IDLE), (IDLE)
- 10) PARAM, (CADEIA, " "), (IDLE), (IDLE)
- 11) WRITE, (INT, 2), (IDLE), (IDLE)
- 12) MAIS, (VAR, temp), (INT, 1), (VAR, ##23)
- 13) ATRIB, (VAR, ##23), (IDLE), (VAR, temp)
- 14) JUMP, (IDLE), (IDLE), (ROTULO, 3)
- 15) NOP, (IDLE), (IDLE), (IDLE)
- 16) IND, (INT, 19), (IDLE), (IDLE)
- 17) INDEX, (VAR, tos), (INT, 1), (VAR, ##26)
- 18) CONTAPONT, (VAR, ##26), (IDLE), (VAR, ##27)
- 19) PARAM, (VAR, ##27), (IDLE), (IDLE)
- 20) WRITE, (INT, 1), (IDLE), (IDLE)
- 21) RETORNO, (IDLE), (IDLE), (IDLE)

Quadruplas do modulo preenchartabelah:

- 1) OPENMOD, (MODULO, preenchartabelah), (IDLE), (IDLE)
- 2) ATRIB, (INT, 0), (IDLE), (VAR, cx)
- 3) NOP, (IDLE), (IDLE), (IDLE)
- 4) LT, (VAR, cx), (INT, 5), (VAR, ##28)
- 5) JF, (VAR, ##28), (IDLE), (ROTULO, 25)
- 6) ATRIB, (INT, 0), (IDLE), (VAR, cy)
- 7) NOP, (IDLE), (IDLE), (IDLE)
- 8) LT, (VAR, cy), (INT, 6), (VAR, ##30)
- 9) JF, (VAR, ##30), (IDLE), (ROTULO, 21)
- 10) IND, (VAR, cx), (IDLE), (IDLE)
- 11) IND, (VAR, cy), (IDLE), (IDLE)
- 12) INDEX, (VAR, h), (INT, 2), (VAR, ##32)
- 13) MULT, (VAR, cx), (VAR, razaox), (VAR, ##33)
- 14) MAIS, (VAR, inicial), (VAR, ##33), (VAR, ##34)
- 15) MULT, (VAR, cy), (VAR, razaoy), (VAR, ##35)
- 16) MAIS, (VAR, ##34), (VAR, ##35), (VAR, ##36)
- 17) ATRIBPONT, (VAR, ##36), (IDLE), (VAR, ##32)
- 18) MAIS, (VAR, cy), (INT, 1), (VAR, ##31)
- 19) ATRIB, (VAR, ##31), (IDLE), (VAR, cy)
- 20) JUMP, (IDLE), (IDLE), (ROTULO, 7)
- 21) NOP, (IDLE), (IDLE), (IDLE)
- 22) MAIS, (VAR, cx), (INT, 1), (VAR, ##29)
- 23) ATRIB, (VAR, ##29), (IDLE), (VAR, cx)
- 24) JUMP, (IDLE), (IDLE), (ROTULO, 3)
- 25) NOP, (IDLE), (IDLE), (IDLE)
- 26) RETORNO, (IDLE), (IDLE), (IDLE)