

Saída dos testes - Compiladores (Analisador Léxico)

Autor: Diego Ascânio Santos

arquivo: test/teste_1.l

```
app testel
  int a, b, c;
  int result
start
  read (a);
  read (c);
  b := 10.23 + a + c;
  result := (a + b) -;
  write(result);
  testel := 1
stop
```

saída: java -jar dist/CompiladorAscanio.jar test/teste_1.l

Sequencia de tokens identificadas:

```
256, app
285, testel
285, int
285, a
44, 44
285, b
44, 44
285, c
59, 59
285, int
285, result
257, start
269, read
40, 40
285, a
41, 41
59, 59
269, read
40, 40
285, c
41, 41
59, 59
285, b
287, :=
284, 10.23
43, 43
285, a
43, 43
285, c
59, 59
285, result
287, :=
40, 40
285, a
43, 43
285, b
41, 41
45, 45
59, 59
270, write
40, 40
285, result
41, 41
59, 59
285, testel
287, :=
283, 1
258, stop
65535, 65535
Símbolos instalados na tabela de símbolos:
44, 44
```

```
1, 1
59, 59
59, 59
40, 40
45, 45
65535, 65535
44, 44
app, app
40, 40
testel, testel
59, 59
write, write
int, int
a, a
41, 41
:=, :=
41, 41
59, 59
43, 43
40, 40
43, 43
59, 59
read, read
40, 40
stop, stop
10.23, 10.23
result, result
59, 59
41, 41
b, b
43, 43
41, 41
c, c
start, start
```

arquivo: test/teste_2.l

```
app
  int a;
  int b;
  real f, _f
start
  read (a);
  read (f);
  f := a * a;
  b := b + a/2 * (3 + 5);
  write(f);
  write(b);
```

saída: java -jar dist/CompiladorAscanio.jar test/teste_2.l

Sequencia de tokens identificadas:

```
256, app
285, int
285, a
59, 59
285, int
285, b
59, 59
260, real
285, f
44, 44
285, _f
257, start
269, read
40, 40
285, a
41, 41
59, 59
269, read
40, 40
285, f
```

```
41, 41
59, 59
285, f
287, :=
285, a
42, 42
285, a
59, 59
285, b
287, :=
285, b
43, 43
285, a
47, 47
283, 2
42, 42
40, 40
283, 3
43, 43
283, 5
41, 41
59, 59
270, write
40, 40
285, f
41, 41
59, 59
270, write
40, 40
285, b
41, 41
59, 59
65535, 65535
Símbolos instalados na tabela de símbolos:
_f, _f
43, 43
b, b
40, 40
41, 41
read, read
44, 44
3, 3
40, 40
47, 47
59, 59
start, start
41, 41
40, 40
41, 41
65535, 65535
59, 59
59, 59
59, 59
42, 42
app, app
59, 59
a, a
40, 40
real, real
:=, :=
59, 59
43, 43
write, write
42, 42
41, 41
int, int
f, f
40, 40
59, 59
59, 59
41, 41
2, 2
```

5, 5

arquivo: test/teste_3.l

```
App pessoa
  int: cont;
  real altura, soma;
start
  cont := 5;
  soma := 0;

  repeat

    write({Altura: });
    read (altura);
    soma := soma # altura;
    cont := cont - 1

  until(cont=0)
  media := soma / 5;

  write({Media: });
  write (media)

stop
```

saída: java -jar dist/CompiladorAscanio.jar test/teste_3.l

Sequencia de tokens identificadas:

```
256, app
285, pessoa
285, int
58, 58
285, cont
59, 59
260, real
285, altura
44, 44
285, soma
59, 59
257, start
285, cont
287, :=
283, 5
59, 59
285, soma
287, :=
283, 0
59, 59
264, repeat
270, write
40, 40
286, Altura: }
125, 125
41, 41
59, 59
269, read
40, 40
285, altura
41, 41
59, 59
285, soma
287, :=
285, soma
35, 35
285, altura
59, 59
285, cont
287, :=
285, cont
45, 45
283, 1
```

```
265, until
40, 40
285, cont
273, =
283, 0
41, 41
285, media
287, :=
285, soma
47, 47
283, 5
59, 59
270, write
40, 40
286, Media: }
125, 125
41, 41
59, 59
270, write
40, 40
285, media
41, 41
258, stop
65535, 65535
Simbolos instalados na tabela de simbolos:
40, 40
soma, soma
125, 125
cont, cont
until, until
59, 59
41, 41
start, start
47, 47
44, 44
41, 41
:=, :=
41, 41
125, 125
59, 59
41, 41
=, =
59, 59
5, 5
stop, stop
40, 40
59, 59
41, 41
59, 59
media, media
45, 45
40, 40
app, app
58, 58
int, int
0, 0
65535, 65535
real, real
write, write
repeat, repeat
59, 59
1, 1
read, read
40, 40
pessoa, pessoa
altura, altura
Media: }, Media: }
0, 0
Altura: }, Altura: }
40, 40
59, 59
59, 59
```

59, 59
5, 5
35, 35

arquivo: test/teste_4.l

App teste4

```
int i, j, k, @total, lsoma, teste4;
float i, a
start
  read (i);
  read (j);
  read (k);
  i := 4 * (5-3 * 50 / 10;
  j := i * 10;
  k := i * j / k;
  k := 4 + a $;
  write(i);
  write(j);
  write(k);
stop
```

saída: java -jar dist/CompiladorAscanio.jar

Sequencia de tokens identificadas:

256, app
285, teste4
285, int
285, i
44, 44
285, j
44, 44
285, k
44, 44
64, 64
285, total
44, 44
283, l
285, soma
44, 44
285, teste4
59, 59
285, float
285, i
44, 44
285, a
257, start
269, read
40, 40
285, i
41, 41
59, 59
269, read
40, 40
285, j
41, 41
59, 59
269, read
40, 40
285, k
41, 41
59, 59
285, i
287, :=
283, 4
42, 42
40, 40
283, 5
45, 45
283, 3
42, 42

```
283, 50
47, 47
283, 10
59, 59
285, j
287, :=
285, i
42, 42
283, 10
59, 59
285, k
287, :=
285, i
42, 42
285, j
47, 47
285, k
59, 59
285, k
287, :=
283, 4
43, 43
285, a
36, 36
59, 59
270, write
40, 40
285, i
41, 41
59, 59
270, write
40, 40
285, j
41, 41
59, 59
270, write
40, 40
285, k
41, 41
59, 59
258, stop
65535, 65535
Simbolos instalados na tabela de simbolos:
59, 59
64, 64
41, 41
40, 40
44, 44
42, 42
float, float
59, 59
44, 44
41, 41
50, 50
44, 44
59, 59
59, 59
59, 59
1, 1
59, 59
10, 10
10, 10
41, 41
5, 5
total, total
soma, soma
4, 4
41, 41
40, 40
42, 42
3, 3
59, 59
```

```
j, j
65535, 65535
42, 42
:=, :=
start, start
app, app
i, i
int, int
write, write
59, 59
44, 44
41, 41
a, a
44, 44
40, 40
4, 4
40, 40
40, 40
teste4, teste4
k, k
40, 40
43, 43
42, 42
36, 36
45, 45
40, 40
read, read
47, 47
44, 44
59, 59
59, 59
59, 59
47, 47
stop, stop
41, 41
```

arquivo: test/test_5.l

app Teste5

```
    int j, k;
    real a, j
start
    read (j);
    read (k);
    if (k != 0)
        result := j/k
    else
        result := 0
    end;

    write(res);
stop
```

saída: java -jar dist/CompiladorAscanio.jar test/teste_5.l

Sequencia de tokens identificadas:

```
256, app
285, teste5
285, int
285, j
44, 44
285, k
59, 59
260, real
285, a
44, 44
285, j
257, start
269, read
40, 40
285, j
```



```
41, 41
59, 59
269, read
40, 40
285, k
41, 41
59, 59
261, if
40, 40
285, k
274, !=
283, 0
41, 41
285, result
287, :=
285, j
47, 47
285, k
263, else
285, result
287, :=
283, 0
266, end
59, 59
270, write
40, 40
285, res
41, 41
59, 59
258, stop
65535, 65535
Simbolos instalados na tabela de simbolos:
```

```
a, a
65535, 65535
!=, !=
j, j
40, 40
40, 40
start, start
real, real
47, 47
read, read
0, 0
41, 41
44, 44
40, 40
:=, :=
stop, stop
44, 44
59, 59
if, if
int, int
teste5, teste5
59, 59
k, k
59, 59
41, 41
else, else
write, write
result, result
res, res
app, app
59, 59
59, 59
40, 40
41, 41
end, end
41, 41
0, 0
```

arquivo: test/teste_6.l

```
%Um programa para calcular a maior idade
int a, b, c, maior;
start
    read(a);
    read(b);
    read(c);

    maior = 0;

    if ( a>b && a>c ) then
        maior := a;
    else
        if ( b > c ) then
            maior := b
        else
            maior := c;
        end
    end

    write({Maior idade: });
    write(maior);
stop
```

saída: java -jar test/teste_6.l

Sequencia de tokens identificadas:

```
10, 10
285, int
285, a
44, 44
285, b
44, 44
285, c
44, 44
285, maior
59, 59
257, start
269, read
40, 40
285, a
41, 41
59, 59
269, read
40, 40
285, b
41, 41
59, 59
269, read
40, 40
285, c
41, 41
59, 59
285, maior
273, =
283, 0
59, 59
261, if
40, 40
285, a
62, 62
285, b
271, &&
285, a
62, 62
285, c
41, 41
262, then
285, maior
287, :=
285, a
59, 59
263, else
```

```
261, if
40, 40
285, b
62, 62
285, c
41, 41
262, then
285, maior
287, :=
285, b
263, else
285, maior
287, :=
285, c
59, 59
266, end
266, end
270, write
40, 40
Exception in thread "main" java.lang.Exception: Erro na linha: 19! Literal mal formado!
    at compiladorascanio.Lexer.scan(Lexer.java:156)
    at compiladorascanio.CompiladorAscanio.main(CompiladorAscanio.java:23)
```

- erro na linha 19, literal sem '}'. Corrigindo, eis a saída:

Sequencia de tokens identificadas:

```
10, 10
285, int
285, a
44, 44
285, b
44, 44
285, c
44, 44
285, maior
59, 59
257, start
269, read
40, 40
285, a
41, 41
59, 59
269, read
40, 40
285, b
41, 41
59, 59
269, read
40, 40
285, c
41, 41
59, 59
285, maior
273, =
283, 0
59, 59
261, if
40, 40
285, a
62, 62
285, b
271, &&
285, a
62, 62
285, c
41, 41
262, then
285, maior
287, :=
285, a
59, 59
263, else
```

```
261, if
40, 40
285, b
62, 62
285, c
41, 41
262, then
285, maior
287, :=
285, b
263, else
285, maior
287, :=
285, c
59, 59
266, end
266, end
270, write
40, 40
286, Maior idade: }
125, 125
41, 41
59, 59
270, write
40, 40
285, maior
41, 41
59, 59
258, stop
65535, 65535
Simbolos instalados na tabela de simbolos:
Maior idade: }, Maior idade: }
59, 59
=, =
b, b
59, 59
41, 41
59, 59
read, read
end, end
maior, maior
62, 62
40, 40
59, 59
41, 41
40, 40
0, 0
40, 40
start, start
41, 41
stop, stop
40, 40
40, 40
41, 41
44, 44
59, 59
then, then
41, 41
10, 10
44, 44
a, a
40, 40
65535, 65535
c, c
40, 40
59, 59
&&, &&
:=, :=
if, if
else, else
int, int
44, 44
```

```
125, 125
62, 62
59, 59
41, 41
59, 59
41, 41
59, 59
write, write
62, 62
```

arquivo: test/teste_7.l

```
app teste_7
  int a, b, ;
read(b)
read(c)
write({a, b})
write({a})
end
```

saída: java -jar dist/CompiladorAscanio.jar test/teste_7.l

Sequencia de tokens identificadas:

```
256, app
285, teste_7
285, int
285, a
44, 44
285, b
44, 44
59, 59
269, read
40, 40
285, b
41, 41
269, read
40, 40
285, c
41, 41
270, write
40, 40
286, a, b}
125, 125
41, 41
270, write
40, 40
```

```
Exception in thread "main" java.lang.Exception: Erro na linha: 6! Literal mal formado!
    at compiladorascanio.Lexer.scan(Lexer.java:156)
    at compiladorascanio.CompiladorAscanio.main(CompiladorAscanio.java:23)
```

- novamente, faltou fechar '}'. Corrigindo o Erro:

Sequencia de tokens identificadas:

```
256, app
285, teste_7
285, int
285, a
44, 44
285, b
44, 44
59, 59
269, read
40, 40
285, b
41, 41
269, read
40, 40
285, c
41, 41
270, write
40, 40
286, a, b}
```

```
125, 125
41, 41
270, write
40, 40
286, a}
125, 125
41, 41
266, end
65535, 65535
Simbolos instalados na tabela de simbolos:
app, app
59, 59
65535, 65535
40, 40
b, b
teste_7, teste_7
a, b}, a, b}
int, int
a}, a}
a, a
44, 44
40, 40
125, 125
end, end
40, 40
41, 41
125, 125
41, 41
c, c
41, 41
write, write
41, 41
44, 44
read, read
40, 40
```

arquivo: test/teste_8.l

```
if ( b >= a )
  c := 8 * (4 - 3;
d:=4
start
write(c)
end
```

saída: java -jar dist/CompiladorAscanio.jar test/teste_8.l

Sequencia de tokens identificadas:

```
261, if
40, 40
285, b
62, 62
276, >=
285, a
41, 41
285, c
287, :=
283, 8
42, 42
40, 40
283, 4
45, 45
283, 3
59, 59
285, d
287, :=
283, 4
257, start
270, write
40, 40
285, c
41, 41
```

```
266, end
65535, 65535
Símbolos instalados na tabela de símbolos:
if, if
c, c
59, 59
a, a
40, 40
d, d
b, b
40, 40
62, 62
>=, >=
write, write
4, 4
8, 8
45, 45
41, 41
start, start
4, 4
42, 42
65535, 65535
3, 3
end, end
41, 41
:=, :=
40, 40
```