

```
1 "C:\Users\Tevin\Documents\Computer Science\Compiler  
Construction\mini-compiler\Paw\paw\Scripts\python.exe  
" "C:\Users\Tevin\Documents\Computer Science\Compiler  
Construction\mini-compiler\Paw\main.py"  
2  
3 INITIATED SYMBOL TABLE global None  
4  
5 FF  
6 /\_\_/  
7 /- -\|  
8 | |__| |  
9 \_\|\_/_  
10 / /\ \\  
11 | / / \ \n\ \ / /  
12 \_\_/_/  
13  
14 Hello, Tevin! Welcome to the Monke compiler!  
15 Type in commands to get started (or 'help' for  
options):  
16 (MonkePaw)> prs  
17 (Parser)> pd  
18 Enter your statement: let a,b,c; a = 10; b = 5; let d  
,e; c=a+4+b;;; d = c + 10;e = a +d*b/c;;;  
19 token found-> <Type:LET, Lexeme:let, BeginPosition:0  
, LinePosition:1>  
20  
21 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
:0, LinePosition:1>])  
22  
23 token found-> <Type:IDENT, Lexeme:a, BeginPosition:4  
, LinePosition:1>  
24  
25 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
:0, LinePosition:1>, <Type:IDENT, Lexeme:a,  
BeginPosition:4, LinePosition:1>])  
26  
27 token found-> <Type:,, Lexeme:,, BeginPosition:5,  
LinePosition:1>  
28  
29 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
:0, LinePosition:1>, <Type:IDENT, Lexeme:a,
```

```
29 BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,  
    BeginPosition:5, LinePosition:1>])  
30  
31 token found-> <Type:IDENT, Lexeme:b, BeginPosition:6  
    , LinePosition:1>  
32  
33 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,  
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme  
    :b, BeginPosition:6, LinePosition:1>])  
34  
35 token found-> <Type:,, Lexeme:,, BeginPosition:7,  
    LinePosition:1>  
36  
37 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,  
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme  
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme  
    :,, BeginPosition:7, LinePosition:1>])  
38  
39 token found-> <Type:IDENT, Lexeme:c, BeginPosition:8  
    , LinePosition:1>  
40  
41 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,  
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme  
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme  
    :,, BeginPosition:7, LinePosition:1>, <Type:IDENT,  
    Lexeme:c, BeginPosition:8, LinePosition:1>])  
42  
43 token found-> <Type:;, Lexeme:;, BeginPosition:9,  
    LinePosition:1>  
44  
45 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,  
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme  
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme
```

```
45 :,, BeginPosition:7, LinePosition:1>, <Type:IDENT,
    Lexeme:c, BeginPosition:8, LinePosition:1>, <Type:;;
    Lexeme:;, BeginPosition:9, LinePosition:1>])
46
47 token found-> <Type:IDENT, Lexeme:a, BeginPosition:11
    , LinePosition:1>
48
49 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,,
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme
    :,, BeginPosition:7, LinePosition:1>, <Type:IDENT,
    Lexeme:c, BeginPosition:8, LinePosition:1>, <Type:;;
    Lexeme:;, BeginPosition:9, LinePosition:1>, <Type:
    IDENT, Lexeme:a, BeginPosition:11, LinePosition:1>])
50
51 token found-> <Type:=, Lexeme:=, BeginPosition:13,
    LinePosition:1>
52
53 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,,
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme
    :,, BeginPosition:7, LinePosition:1>, <Type:IDENT,
    Lexeme:c, BeginPosition:8, LinePosition:1>, <Type:;;
    Lexeme:;, BeginPosition:9, LinePosition:1>, <Type:
    IDENT, Lexeme:a, BeginPosition:11, LinePosition:1>, <
    Type:=, Lexeme:=, BeginPosition:13, LinePosition:1>])
54
55 token found-> <Type:INT, Lexeme:10, BeginPosition:15
    , LinePosition:1>
56
57 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,,
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme
    :,, BeginPosition:7, LinePosition:1>, <Type:IDENT,
    Lexeme:c, BeginPosition:8, LinePosition:1>, <Type:;;,
```

```
57 Lexeme:;, BeginPosition:9, LinePosition:1>, <Type:  
    IDENT, Lexeme:a, BeginPosition:11, LinePosition:1>, <  
    Type:=, Lexeme:=, BeginPosition:13, LinePosition:1  
>, <Type:INT, Lexeme:10, BeginPosition:15,  
    LinePosition:1>])  
58  
59 token found-> <Type:;, Lexeme:;, BeginPosition:17,  
    LinePosition:1>  
60  
61 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,  
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme  
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme  
    :,, BeginPosition:7, LinePosition:1>, <Type:IDENT,  
    Lexeme:c, BeginPosition:8, LinePosition:1>, <Type:;,  
    Lexeme:;, BeginPosition:9, LinePosition:1>, <Type:  
    IDENT, Lexeme:a, BeginPosition:11, LinePosition:1>, <  
    Type:=, Lexeme:=, BeginPosition:13, LinePosition:1  
>, <Type:INT, Lexeme:10, BeginPosition:15,  
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:17  
    , LinePosition:1>])  
62  
63 token found-> <Type:IDENT, Lexeme:b, BeginPosition:19  
    , LinePosition:1>  
64  
65 tokens -> deque([<Type:LET, Lexeme:let, BeginPosition  
    :0, LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:4, LinePosition:1>, <Type:,, Lexeme:,,  
    BeginPosition:5, LinePosition:1>, <Type:IDENT, Lexeme  
    :b, BeginPosition:6, LinePosition:1>, <Type:,, Lexeme  
    :,, BeginPosition:7, LinePosition:1>, <Type:IDENT,  
    Lexeme:c, BeginPosition:8, LinePosition:1>, <Type:;,  
    Lexeme:;, BeginPosition:9, LinePosition:1>, <Type:  
    IDENT, Lexeme:a, BeginPosition:11, LinePosition:1>, <  
    Type:=, Lexeme:=, BeginPosition:13, LinePosition:1  
>, <Type:INT, Lexeme:10, BeginPosition:15,  
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:17  
    , LinePosition:1>, <Type:IDENT, Lexeme:b,  
    BeginPosition:19, LinePosition:1>])  
66
```

```
67 token found-> <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>
68
69 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>])
70
71 token found-> <Type:INT, Lexeme:5, BeginPosition:23
    , LinePosition:1>
72
73 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>])
74
```

```
75 token found-> <Type:;, Lexeme:;, BeginPosition:24,  
LinePosition:1>  
76  
77 tokens -> deque([<Type:LET, Lexeme:let,  
BeginPosition:0, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:  
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition  
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,  
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9  
, LinePosition:1>, <Type:IDENT, Lexeme:a,  
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:13, LinePosition:1>, <Type:INT,  
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type  
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition  
:1>, <Type:=, Lexeme:=, BeginPosition:21,  
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:  
23, LinePosition:1>, <Type:;, Lexeme:;,  
BeginPosition:24, LinePosition:1>])  
78  
79 token found-> <Type:LET, Lexeme:let, BeginPosition:  
26, LinePosition:1>  
80  
81 tokens -> deque([<Type:LET, Lexeme:let,  
BeginPosition:0, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:  
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition  
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,  
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9  
, LinePosition:1>, <Type:IDENT, Lexeme:a,  
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:13, LinePosition:1>, <Type:INT,  
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type  
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition  
:1>, <Type:=, Lexeme:=, BeginPosition:21,  
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
```

```
81 23, LinePosition:1>, <Type:;, Lexeme:;,  
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme  
    :let, BeginPosition:26, LinePosition:1>])  
82  
83 token found-> <Type:IDENT, Lexeme:d, BeginPosition:  
    30, LinePosition:1>  
84  
85 tokens -> deque([<Type:LET, Lexeme:let,  
    BeginPosition:0, LinePosition:1>, <Type:IDENT,  
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
    :;, Lexeme:;, BeginPosition:5, LinePosition:1>, <  
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:  
    1>, <Type:;, Lexeme:;, BeginPosition:7, LinePosition  
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,  
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9  
    , LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme  
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,  
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type  
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <  
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition  
    :1>, <Type:=, Lexeme:=, BeginPosition:21,  
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:  
    23, LinePosition:1>, <Type:;, Lexeme:;,  
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme  
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT  
    , Lexeme:d, BeginPosition:30, LinePosition:1>])  
86  
87 token found-> <Type:;, Lexeme:;, BeginPosition:31,  
    LinePosition:1>  
88  
89 tokens -> deque([<Type:LET, Lexeme:let,  
    BeginPosition:0, LinePosition:1>, <Type:IDENT,  
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
    :;, Lexeme:;, BeginPosition:5, LinePosition:1>, <  
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:  
    1>, <Type:;, Lexeme:;, BeginPosition:7, LinePosition  
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,  
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9  
    , LinePosition:1>, <Type:IDENT, Lexeme:a,  
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
```

```
89 :=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
::, Lexeme::, BeginPosition:17, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
:1>, <Type:=, Lexeme:=, BeginPosition:21,
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
23, LinePosition:1>, <Type::, Lexeme::,
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
::, Lexeme::, BeginPosition:31, LinePosition:1>])
90
91 token found-> <Type:IDENT, Lexeme:e, BeginPosition:
32, LinePosition:1>
92
93 tokens -> deque([<Type:LET, Lexeme:let,
BeginPosition:0, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
::, Lexeme::, BeginPosition:5, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
1>, <Type::, Lexeme::, BeginPosition:7, LinePosition
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
LinePosition:1>, <Type::, Lexeme::, BeginPosition:9
, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
::, Lexeme::, BeginPosition:17, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
:1>, <Type:=, Lexeme:=, BeginPosition:21,
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
23, LinePosition:1>, <Type::, Lexeme::,
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
::, Lexeme::, BeginPosition:31, LinePosition:1>, <
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
:1>])
94
95 token found-> <Type::, Lexeme::, BeginPosition:33,
LinePosition:1>
```

```
96
97 tokens -> deque([<Type:LET, Lexeme:let,
BeginPosition:0, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
:1>, <Type:=, Lexeme:=, BeginPosition:21,
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
23, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
:1>, <Type:;, Lexeme:;, BeginPosition:33,
LinePosition:1>])
98
99 token found-> <Type:IDENT, Lexeme:c, BeginPosition:
35, LinePosition:1>
100
101 tokens -> deque([<Type:LET, Lexeme:let,
BeginPosition:0, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
```

```
101 :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>])  
102  
103 token found-> <Type:=, Lexeme:=, BeginPosition:36,
    LinePosition:1>  
104  
105 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
```

```
105 BeginPosition:35, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:36, LinePosition:1>])  
106  
107 token found-> <Type:IDENT, Lexeme:a, BeginPosition:  
37, LinePosition:1>  
108  
109 tokens -> deque([<Type:LET, Lexeme:let,  
BeginPosition:0, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:  
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition  
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,  
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9  
, LinePosition:1>, <Type:IDENT, Lexeme:a,  
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:13, LinePosition:1>, <Type:INT,  
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type  
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition  
:1>, <Type:=, Lexeme:=, BeginPosition:21,  
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:  
23, LinePosition:1>, <Type:;, Lexeme:;,  
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme  
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT  
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type  
:,, Lexeme:,, BeginPosition:31, LinePosition:1>, <  
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition  
:1>, <Type:;, Lexeme:;, BeginPosition:33,  
LinePosition:1>, <Type:IDENT, Lexeme:c,  
BeginPosition:35, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:36, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:37, LinePosition:1>])  
110  
111 token found-> <Type:+, Lexeme:+, BeginPosition:38,  
LinePosition:1>  
112  
113 tokens -> deque([<Type:LET, Lexeme:let,  
BeginPosition:0, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
```

```

113 Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>])

114

115 token found-> <Type:INT, Lexeme:4, BeginPosition:39
    , LinePosition:1>

116

117 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <

```

```
117 Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >])
118
119 token found-> <Type:+, Lexeme:+, BeginPosition:40,
    LinePosition:1>
120
121 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
```

```
121 :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>])
122
123 token found-> <Type:IDENT, Lexeme:b, BeginPosition:
    41, LinePosition:1>
124
125 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
```

```
125 Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>])
126
127 token found-> <Type:;, Lexeme:;, BeginPosition:42,
    LinePosition:1>
128
129 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
```

```
129 LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>])
130
131 token found-> <Type:;, Lexeme:;, BeginPosition:43,
    LinePosition:1>
132
133 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <Type
    :TYPE:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>])
134
```

```
135 token found-> <Type:;, Lexeme:;, BeginPosition:44,  
LinePosition:1>  
136  
137 tokens -> deque([<Type:LET, Lexeme:let,  
BeginPosition:0, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
:;, Lexeme:;, BeginPosition:5, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:  
1>, <Type:;, Lexeme:;, BeginPosition:7, LinePosition  
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,  
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9  
, LinePosition:1>, <Type:IDENT, Lexeme:a,  
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:13, LinePosition:1>, <Type:INT,  
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type  
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition  
:1>, <Type:=, Lexeme:=, BeginPosition:21,  
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:  
23, LinePosition:1>, <Type:;, Lexeme:;,  
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme  
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT  
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type  
:;, Lexeme:;, BeginPosition:31, LinePosition:1>, <  
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition  
:1>, <Type:;, Lexeme:;, BeginPosition:33,  
LinePosition:1>, <Type:IDENT, Lexeme:c,  
BeginPosition:35, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:36, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:37, LinePosition:1>, <Type  
:+, Lexeme:+, BeginPosition:38, LinePosition:1>, <  
Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1  
>, <Type:+, Lexeme:+, BeginPosition:40, LinePosition  
:1>, <Type:IDENT, Lexeme:b, BeginPosition:41,  
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42  
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:  
43, LinePosition:1>, <Type:;, Lexeme:;,  
BeginPosition:44, LinePosition:1>])  
138  
139 token found-> <Type:IDENT, Lexeme:d, BeginPosition:  
46, LinePosition:1>
```

```
140
141 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>])
142
143 token found-> <Type:=, Lexeme:=, BeginPosition:48,
    LinePosition:1>
144
```

```
145 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>])
146
147 token found-> <Type:IDENT, Lexeme:c, BeginPosition:
    50, LinePosition:1>
148
```

```
149 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>])
```

150

151 token found-> <Type:+, Lexeme:+, BeginPosition:52,

```
151 LinePosition:1>
152
153 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type::, Lexeme::, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type::, Lexeme::, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type::, Lexeme::,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type::, Lexeme::, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    ::+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type::, Lexeme::, BeginPosition:42
    , LinePosition:1>, <Type::, Lexeme::, BeginPosition:
    43, LinePosition:1>, <Type::, Lexeme::,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
```

```
153 LinePosition:1>])
154
155 token found-> <Type:INT, Lexeme:10, BeginPosition:54
, LinePosition:1>
156
157 tokens -> deque([<Type:LET, Lexeme:let,
BeginPosition:0, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
:1>, <Type:=, Lexeme:=, BeginPosition:21,
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
23, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
:1>, <Type:;, Lexeme:;, BeginPosition:33,
LinePosition:1>, <Type:IDENT, Lexeme:c,
BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
:+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
>, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
:1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
43, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:44, LinePosition:1>, <Type:IDENT,
Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
```

```
157 :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>])
158
159 token found-> <Type:;, Lexeme:;, BeginPosition:56,
    LinePosition:1>
160
161 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;;
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
```

```
161 , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:  
43, LinePosition:1>, <Type:;, Lexeme:;,  
BeginPosition:44, LinePosition:1>, <Type:IDENT,  
Lexeme:d, BeginPosition:46, LinePosition:1>, <Type  
:=, Lexeme:=, BeginPosition:48, LinePosition:1>, <  
Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition  
:1>, <Type:+, Lexeme:+, BeginPosition:52,  
LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition  
:54, LinePosition:1>, <Type:;, Lexeme:;,  
BeginPosition:56, LinePosition:1>])  
162  
163 token found-> <Type:IDENT, Lexeme:e, BeginPosition:  
57, LinePosition:1>  
164  
165 tokens -> deque([<Type:LET, Lexeme:let,  
BeginPosition:0, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type  
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:  
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition  
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,  
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9  
, LinePosition:1>, <Type:IDENT, Lexeme:a,  
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:13, LinePosition:1>, <Type:INT,  
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type  
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <  
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition  
:1>, <Type:=, Lexeme:=, BeginPosition:21,  
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:  
23, LinePosition:1>, <Type:;, Lexeme:;,  
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme  
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT  
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type  
:,, Lexeme:,, BeginPosition:31, LinePosition:1>, <  
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition  
:1>, <Type:;, Lexeme:;, BeginPosition:33,  
LinePosition:1>, <Type:IDENT, Lexeme:c,  
BeginPosition:35, LinePosition:1>, <Type:=, Lexeme  
:=, BeginPosition:36, LinePosition:1>, <Type:IDENT,  
Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
```

```

165 :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>])

166

167 token found-> <Type:=, Lexeme:=, BeginPosition:59,
    LinePosition:1>

168

169 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <

```

```

169 Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    42, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>])
170
171 token found-> <Type:IDENT, Lexeme:a, BeginPosition:
    61, LinePosition:1>
172
173 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>])

```

```

173 :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;, 
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
    Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
    :1>])
174
175 token found-> <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>
176
177 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:

```

```
177 1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
:1>, <Type:=, Lexeme:=, BeginPosition:21,
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
23, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
:1>, <Type:;, Lexeme:;, BeginPosition:33,
LinePosition:1>, <Type:IDENT, Lexeme:c,
BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
:+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
>, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
:1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
43, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:44, LinePosition:1>, <Type:IDENT,
Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
:=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
:1>, <Type:+, Lexeme:+, BeginPosition:52,
LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
:54, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:56, LinePosition:1>, <Type:IDENT,
Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
:=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
:1>, <Type:+, Lexeme:+, BeginPosition:63,
```

```
177 LinePosition:1>])
178
179 token found-> <Type:IDENT, Lexeme:d, BeginPosition:
   64, LinePosition:1>
180
181 tokens -> deque([<Type:LET, Lexeme:let,
   BeginPosition:0, LinePosition:1>, <Type:IDENT,
   Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
   ::, Lexeme::, BeginPosition:5, LinePosition:1>, <
   Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
   1>, <Type::, Lexeme::, BeginPosition:7, LinePosition
   :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
   LinePosition:1>, <Type::, Lexeme::, BeginPosition:9
   , LinePosition:1>, <Type:IDENT, Lexeme:a,
   BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
   :=, BeginPosition:13, LinePosition:1>, <Type:INT,
   Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
   ::, Lexeme::, BeginPosition:17, LinePosition:1>, <
   Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
   :1>, <Type:=, Lexeme:=, BeginPosition:21,
   LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
   23, LinePosition:1>, <Type::, Lexeme::,
   BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
   :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
   , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
   ::, Lexeme::, BeginPosition:31, LinePosition:1>, <
   Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
   :1>, <Type::, Lexeme::, BeginPosition:33,
   LinePosition:1>, <Type:IDENT, Lexeme:c,
   BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
   :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
   Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
   ::+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
   Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
   >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
   :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
   LinePosition:1>, <Type::, Lexeme::, BeginPosition:42
   , LinePosition:1>, <Type::, Lexeme::, BeginPosition:
   43, LinePosition:1>, <Type::, Lexeme::,
   BeginPosition:44, LinePosition:1>, <Type:IDENT,
   Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
```

```

181 :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:;, 
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
    Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>, <Type:IDENT, Lexeme:d,
    BeginPosition:64, LinePosition:1>])

182

183 token found-> <Type:*, Lexeme:*, BeginPosition:65,
    LinePosition:1>

184

185 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;, 
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme

```

```

185 :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
:+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
>, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
:1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
43, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:44, LinePosition:1>, <Type:IDENT,
Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
:=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
:1>, <Type:+, Lexeme:+, BeginPosition:52,
LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
:54, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:56, LinePosition:1>, <Type:IDENT,
Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
:=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
:1>, <Type:+, Lexeme:+, BeginPosition:63,
LinePosition:1>, <Type:IDENT, Lexeme:d,
BeginPosition:64, LinePosition:1>, <Type:*, Lexeme
:*, BeginPosition:65, LinePosition:1>])

186

187 token found-> <Type:IDENT, Lexeme:b, BeginPosition:
66, LinePosition:1>

188

189 tokens -> deque([<Type:LET, Lexeme:let,
BeginPosition:0, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <

```

```
189 Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    42, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
    Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>, <Type:IDENT, Lexeme:d,
    BeginPosition:64, LinePosition:1>, <Type:*, Lexeme
    :*, BeginPosition:65, LinePosition:1>, <Type:IDENT,
    Lexeme:b, BeginPosition:66, LinePosition:1>])
190
191 token found-> <Type:/, Lexeme:/, BeginPosition:67,
    LinePosition:1>
192
```

```
193 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:|,
```

```

193 BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
    Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>, <Type:IDENT, Lexeme:d,
    BeginPosition:64, LinePosition:1>, <Type:*, Lexeme
    :*, BeginPosition:65, LinePosition:1>, <Type:IDENT,
    Lexeme:b, BeginPosition:66, LinePosition:1>, <Type
    :/, Lexeme:/, BeginPosition:67, LinePosition:1>])
194
195 token found-> <Type:IDENT, Lexeme:c, BeginPosition:
    68, LinePosition:1>
196
197 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type::, Lexeme::, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type::, Lexeme::, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type::, Lexeme::,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type::, Lexeme::, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type

```

```

197 :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
    , LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
    43, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
    Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>, <Type:IDENT, Lexeme:d,
    BeginPosition:64, LinePosition:1>, <Type:*, Lexeme
    :*, BeginPosition:65, LinePosition:1>, <Type:IDENT,
    Lexeme:b, BeginPosition:66, LinePosition:1>, <Type
    :/, Lexeme:/, BeginPosition:67, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:68, LinePosition
    :1>])
198
199 token found-> <Type:;, Lexeme:;, BeginPosition:69,
    LinePosition:1>
200
201 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,

```

```
201 Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type::, Lexeme::,
    BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
    :let, BeginPosition:26, LinePosition:1>, <Type:IDENT
    , Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
    ::, Lexeme::, BeginPosition:31, LinePosition:1>, <
    Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
    :1>, <Type::, Lexeme::, BeginPosition:33,
    LinePosition:1>, <Type:IDENT, Lexeme:c,
    BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
    Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
    >, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
    :1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
    LinePosition:1>, <Type::, Lexeme::, BeginPosition:42
    , LinePosition:1>, <Type::, Lexeme::, BeginPosition:
    43, LinePosition:1>, <Type::, Lexeme::,
    BeginPosition:44, LinePosition:1>, <Type:IDENT,
    Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type::, Lexeme::,
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
    Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>, <Type:IDENT, Lexeme:d,
    BeginPosition:64, LinePosition:1>, <Type:*, Lexeme
    :*, BeginPosition:65, LinePosition:1>, <Type:IDENT,
    Lexeme:b, BeginPosition:66, LinePosition:1>, <Type
    :/, Lexeme:/, BeginPosition:67, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:68, LinePosition
```

```
201 :1>, <Type:;, Lexeme:;, BeginPosition:69,
LinePosition:1>])
202
203 token found-> <Type:;, Lexeme:;, BeginPosition:70,
LinePosition:1>
204
205 tokens -> deque([<Type:LET, Lexeme:let,
BeginPosition:0, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:5, LinePosition:1>, <Type
:TYPE:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
:1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:13, LinePosition:1>, <Type:INT,
Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
:;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
:1>, <Type:=, Lexeme:=, BeginPosition:21,
LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
23, LinePosition:1>, <Type:;, Lexeme:;,
BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
:1>, <Type:;, Lexeme:;, BeginPosition:33,
LinePosition:1>, <Type:IDENT, Lexeme:c,
BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
:+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
>, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
:1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
43, LinePosition:1>, <Type:;, Lexeme:;,
BeginPosition:44, LinePosition:1>, <Type:IDENT,
```

```

205 Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:52,
    LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
    :54, LinePosition:1>, <Type:;, Lexeme:|,
    BeginPosition:56, LinePosition:1>, <Type:IDENT,
    Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
    :=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
    Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
    :1>, <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>, <Type:IDENT, Lexeme:d,
    BeginPosition:64, LinePosition:1>, <Type:*, Lexeme
    :*, BeginPosition:65, LinePosition:1>, <Type:IDENT,
    Lexeme:b, BeginPosition:66, LinePosition:1>, <Type
    :/, Lexeme:/, BeginPosition:67, LinePosition:1>, <
    Type:IDENT, Lexeme:c, BeginPosition:68, LinePosition
    :1>, <Type:;, Lexeme:;, BeginPosition:69,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:70
    , LinePosition:1>])
206
207 token found-> <Type:;, Lexeme:;, BeginPosition:71,
    LinePosition:1>
208
209 tokens -> deque([<Type:LET, Lexeme:let,
    BeginPosition:0, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:4, LinePosition:1>, <Type
    :,, Lexeme:,, BeginPosition:5, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition:
    1>, <Type:,, Lexeme:,, BeginPosition:7, LinePosition
    :1>, <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:9
    , LinePosition:1>, <Type:IDENT, Lexeme:a,
    BeginPosition:11, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:13, LinePosition:1>, <Type:INT,
    Lexeme:10, BeginPosition:15, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:17, LinePosition:1>, <
    Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition
    :1>, <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>, <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>, <Type:;, Lexeme:;,
```

```
209 BeginPosition:24, LinePosition:1>, <Type:LET, Lexeme
:let, BeginPosition:26, LinePosition:1>, <Type:IDENT
, Lexeme:d, BeginPosition:30, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:31, LinePosition:1>, <
Type:IDENT, Lexeme:e, BeginPosition:32, LinePosition
:1>, <Type:;, Lexeme:;, BeginPosition:33,
LinePosition:1>, <Type:IDENT, Lexeme:c,
BeginPosition:35, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:36, LinePosition:1>, <Type:IDENT,
Lexeme:a, BeginPosition:37, LinePosition:1>, <Type
:+, Lexeme:+, BeginPosition:38, LinePosition:1>, <
Type:INT, Lexeme:4, BeginPosition:39, LinePosition:1
>, <Type:+, Lexeme:+, BeginPosition:40, LinePosition
:1>, <Type:IDENT, Lexeme:b, BeginPosition:41,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:42
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
43, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:44, LinePosition:1>, <Type:IDENT,
Lexeme:d, BeginPosition:46, LinePosition:1>, <Type
:=, Lexeme:=, BeginPosition:48, LinePosition:1>, <
Type:IDENT, Lexeme:c, BeginPosition:50, LinePosition
:1>, <Type:+, Lexeme:+, BeginPosition:52,
LinePosition:1>, <Type:INT, Lexeme:10, BeginPosition
:54, LinePosition:1>, <Type:;, Lexeme:|,
BeginPosition:56, LinePosition:1>, <Type:IDENT,
Lexeme:e, BeginPosition:57, LinePosition:1>, <Type
:=, Lexeme:=, BeginPosition:59, LinePosition:1>, <
Type:IDENT, Lexeme:a, BeginPosition:61, LinePosition
:1>, <Type:+, Lexeme:+, BeginPosition:63,
LinePosition:1>, <Type:IDENT, Lexeme:d,
BeginPosition:64, LinePosition:1>, <Type:*, Lexeme
:*, BeginPosition:65, LinePosition:1>, <Type:IDENT,
Lexeme:b, BeginPosition:66, LinePosition:1>, <Type
:/, Lexeme:/, BeginPosition:67, LinePosition:1>, <
Type:IDENT, Lexeme:c, BeginPosition:68, LinePosition
:1>, <Type:;, Lexeme:;, BeginPosition:69,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:70
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
71, LinePosition:1>])
```

210

211 Total runtime is 0.00614595

```
212
213 TOKENS - deque([<Type:LET, Lexeme:let, BeginPosition
:0, LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:4, LinePosition:1>, <Type:,, Lexeme
:,, BeginPosition:5, LinePosition:1>, <Type:IDENT,
Lexeme:b, BeginPosition:6, LinePosition:1>, <Type
:,, Lexeme:,, BeginPosition:7, LinePosition:1>, <
Type:IDENT, Lexeme:c, BeginPosition:8, LinePosition:
1>, <Type:;, Lexeme:;, BeginPosition:9, LinePosition
:1>, <Type:IDENT, Lexeme:a, BeginPosition:11,
LinePosition:1>, <Type:=, Lexeme:=, BeginPosition:13
, LinePosition:1>, <Type:INT, Lexeme:10,
BeginPosition:15, LinePosition:1>, <Type:;, Lexeme
:;, BeginPosition:17, LinePosition:1>, <Type:IDENT,
Lexeme:b, BeginPosition:19, LinePosition:1>, <Type
:=, Lexeme:=, BeginPosition:21, LinePosition:1>, <
Type:INT, Lexeme:5, BeginPosition:23, LinePosition:1
>, <Type:;, Lexeme:;, BeginPosition:24, LinePosition
:1>, <Type:LET, Lexeme:let, BeginPosition:26,
LinePosition:1>, <Type:IDENT, Lexeme:d,
BeginPosition:30, LinePosition:1>, <Type:,, Lexeme
:,, BeginPosition:31, LinePosition:1>, <Type:IDENT,
Lexeme:e, BeginPosition:32, LinePosition:1>, <Type
:;, Lexeme:;, BeginPosition:33, LinePosition:1>, <
Type:IDENT, Lexeme:c, BeginPosition:35, LinePosition
:1>, <Type:=, Lexeme:=, BeginPosition:36,
LinePosition:1>, <Type:IDENT, Lexeme:a,
BeginPosition:37, LinePosition:1>, <Type:+, Lexeme
:+, BeginPosition:38, LinePosition:1>, <Type:INT,
Lexeme:4, BeginPosition:39, LinePosition:1>, <Type
:+, Lexeme:+, BeginPosition:40, LinePosition:1>, <
Type:IDENT, Lexeme:b, BeginPosition:41, LinePosition
:1>, <Type:;, Lexeme:;, BeginPosition:42,
LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:43
, LinePosition:1>, <Type:;, Lexeme:;, BeginPosition:
44, LinePosition:1>, <Type:IDENT, Lexeme:d,
BeginPosition:46, LinePosition:1>, <Type:=, Lexeme
:=, BeginPosition:48, LinePosition:1>, <Type:IDENT,
Lexeme:c, BeginPosition:50, LinePosition:1>, <Type
:+, Lexeme:+, BeginPosition:52, LinePosition:1>, <
Type:INT, Lexeme:10, BeginPosition:54, LinePosition:
```

```
213 1>, <Type:;, Lexeme:;, BeginPosition:56,
    LinePosition:1>, <Type:IDENT, Lexeme:e,
    BeginPosition:57, LinePosition:1>, <Type:=, Lexeme
    :=, BeginPosition:59, LinePosition:1>, <Type:IDENT,
    Lexeme:a, BeginPosition:61, LinePosition:1>, <Type
    :+, Lexeme:+, BeginPosition:63, LinePosition:1>, <
    Type:IDENT, Lexeme:d, BeginPosition:64, LinePosition
    :1>, <Type:*, Lexeme:*, BeginPosition:65,
    LinePosition:1>, <Type:IDENT, Lexeme:b,
    BeginPosition:66, LinePosition:1>, <Type:/, Lexeme
    :/, BeginPosition:67, LinePosition:1>, <Type:IDENT,
    Lexeme:c, BeginPosition:68, LinePosition:1>, <Type
    :;, Lexeme:;, BeginPosition:69, LinePosition:1>, <
    Type:;, Lexeme:;, BeginPosition:70, LinePosition:1
    >, <Type:;, Lexeme:;, BeginPosition:71, LinePosition
    :1>, <Type:EOF, Lexeme:, BeginPosition:72,
    LinePosition:1>])
214
215 ProgramNode (type::= 'PROGRAM', name::= '
    Monke_beta_0.1', value:: ='StatementListNode (type
    ::= 'STATEMENTS', name::= 'STATEMENTS', value:: ='de
    que([])'))
216
217 <class 'parser.LL1.ProgramNode'>
218
219 True
220
221
222 Creating a symbol from Node
223 ProgramNode (type::= 'PROGRAM', name::= '
    Monke_beta_0.1', value:: ='StatementListNode (type
    ::= 'STATEMENTS', name::= 'STATEMENTS', value:: ='de
    que([])'))
224 ProgramNode
225
226
227 Creating a symbol from NodeValue
228 StatementListNode (type::= 'STATEMENTS', name::= '
    STATEMENTS', value:: ='deque([])')
229
230
```

```
231 BEFORE DEFINE _contexts and current_context
232 deque([{}])
233 {}
234
235
236 BEFORE DEFINE self: global
237
238 +-----+-----+-----+
239 | Name    | Type     | Line Declared   | Context Level
239 |          | Value     |                  |
240 +=====+=====+=====+=====+=====+
241 |-----+-----+-----+
242
243
244 SAVING NAME: PROGRAM
245
246
247 SAVING SYMBOL: Symbol 'Monke_beta_0.1' of type 'PROGRAM' at context level 0
248
249
250 SAVING CONTEXT: {}
251
252
253 THIS IS CURRENT CONTEXT => {}
254
255
256 Updated Symbol table global
257 +-----+-----+-----+
258 | Name           | Type     | Line Declared   |
258 | Context Level | Value    |
259 +=====+=====+=====+=====+
260 | Monke_beta_0.1 | None     | None:None
```

```
260 | 0 | StatementListNode (type
::= 'STATEMENTS', name::= |
261 | | |
| 'STATEMENTS', value:: = 'deque
([])' |
262 +-----+-----+-----+
-----+-----+
-----+
263
264
265 Parsing <Type:LET, Lexeme:let, BeginPosition:0,
LinePosition:1>
266 Next is <Type:IDENT, Lexeme:a, BeginPosition:4,
LinePosition:1>
267
268
269 Dealing with factor: <Type:IDENT, Lexeme:a,
BeginPosition:4, LinePosition:1>
270 Next dealt terminal/non-term-> token: <Type:,,,
Lexeme:,, BeginPosition:5, LinePosition:1>
271
272 IdentifierNode (type::= 'IDENT', name::= 'a', value
:: = 'None')
273
274 <class 'parser.LL1.IdentifierNode'>
275
276 True
277
278
279 Creating a symbol from Node
280 IdentifierNode (type::= 'IDENT', name::= 'a', value
:: = 'None')
281 IdentifierNode
282
283
284 Creating a symbol from Node.Value.Token
285 <Type:IDENT, Lexeme:a, BeginPosition:4,
LinePosition:1>
286
287
288 BEFORE DEFINE _contexts and current_context
```

```

289 deque([{'PROGRAM': Symbol(name='Monke_beta_0.1',
    type_='PROGRAM', context_level=0)}])
290 {'PROGRAM': Symbol(name='Monke_beta_0.1', type_='
    PROGRAM', context_level=0)}
291
292
293 BEFORE DEFINE self: global
294
295 +-----+-----+-----+
-----+
-----+
296 | Name          | Type      | Line Declared   |
Context Level | Value
|
297 +=====+=====+=====+=====+
=====+=====+=====+=====+
=====+
298 | Monke_beta_0.1 | None     | None:None
|                 | 0 | StatementListNode (type
| ::= 'STATEMENTS', name::= |
299 |                 |           |
|           | 'STATEMENTS', value::: ='dequeue
| ([])'           |
300 +-----+-----+-----+
-----+
-----+
301
302
303 SAVING NAME: a
304
305
306 SAVING SYMBOL: Symbol 'a' of type 'IDENT' at context
level 0
307
308
309 SAVING CONTEXT: {'PROGRAM': Symbol(name='
    Monke_beta_0.1', type_='PROGRAM', context_level=0)}
310
311
312 THIS IS CURRENT CONTEXT => {'PROGRAM': Symbol(name='
    Monke_beta_0.1', type_='PROGRAM', context_level=0)}

```

```

313
314
315 Updated Symbol table global
316 +-----+-----+-----+
-----+
-----+
317 | Name          | Type     | Line Declared   |
Context Level | Value
|
318 +-----+-----+-----+-----+
-----+
-----+
319 | Monke_beta_0.1 | None    | None:None
|                   0 | StatementListNode (type
 ::= 'STATEMENTS', name::= |
320 |                   |           |
|                   | 'STATEMENTS', value:: = 'deque
([[]))'           |
321 +-----+-----+-----+
-----+
-----+
322 | a            | None    | 1:4
|                   0 | None
|
323 +-----+-----+-----+
-----+
-----+
324
325
326 CHILD deque([IdentifierNode (type::= 'IDENT', name
 ::= 'a', value:: = 'None')])
327
328
329 <Type:IDENT, Lexeme:a, BeginPosition:4, LinePosition
 :1> <Type:,, Lexeme:,, BeginPosition:5, LinePosition
 :1>
330
331
332 <Type:,, Lexeme:,, BeginPosition:5, LinePosition:1
 > <Type:IDENT, Lexeme:b, BeginPosition:6,
LinePosition:1>

```

```
333
334
335 IN COMMA
336 <Type:,, Lexeme:,, BeginPosition:5, LinePosition:1
     > <Type:IDENT, Lexeme:b, BeginPosition:6,
        LinePosition:1>
337
338
339 IN COMMA
340 <Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition
   :1> <Type:,, Lexeme:,, BeginPosition:7, LinePosition
   :1>
341
342
343 Dealing with factor: <Type:IDENT, Lexeme:b,
   BeginPosition:6, LinePosition:1>
344 Next dealt terminal/non-term-> token: <Type:,,,
   Lexeme:,, BeginPosition:7, LinePosition:1>
345
346 IdentifierNode (type::= 'IDENT', name::= 'b', value
   :: ='None')
347
348 <class 'parser.LL1.IdentifierNode'>
349
350 True
351
352
353 Creating a symbol from Node
354 IdentifierNode (type::= 'IDENT', name::= 'b', value
   :: ='None')
355 IdentifierNode
356
357
358 Creating a symbol from Node.Value.Token
359 <Type:IDENT, Lexeme:b, BeginPosition:6,
   LinePosition:1>
360
361
362 BEFORE DEFINE _contexts and current_context
363 deque([{'PROGRAM': Symbol(name='Monke_beta_0.1',
   type_='PROGRAM', context_level=0), 'a': Symbol(name
```

```

363 ='a', type_='IDENT', context_level=0)})])
364 {'PROGRAM': Symbol(name='Monke_beta_0.1', type_='
    PROGRAM', context_level=0), 'a': Symbol(name='a',
    type_='IDENT', context_level=0)}
365
366
367 BEFORE DEFINE self: global
368
369 +-----+-----+-----+
-----+
-----+
370 | Name          | Type      | Line Declared   |
Context Level | Value
|
371 +=====+=====+=====+=====+
=====+=====+=====+=====+
=====+
372 | Monke_beta_0.1 | None     | None:None
|                 | 0 | StatementListNode (type
| ::= 'STATEMENTS', name::= |
373 |                 |           |
|           | 'STATEMENTS', value::= 'deque
([])'
374 +-----+-----+-----+
-----+
-----+
375 | a           | None     | 1:4
|           | 0 | None
|
376 +-----+-----+-----+
-----+
-----+
377
378
379 SAVING NAME: b
380
381
382 SAVING SYMBOL: Symbol 'b' of type 'IDENT' at context
level 0
383
384

```

```

385 SAVING CONTEXT: {'PROGRAM': Symbol(name='
    Monke_beta_0.1', type_='PROGRAM', context_level=0
), 'a': Symbol(name='a', type_='IDENT',
    context_level=0)}
386
387
388 THIS IS CURRENT CONTEXT => {'PROGRAM': Symbol(name='
    Monke_beta_0.1', type_='PROGRAM', context_level=0
), 'a': Symbol(name='a', type_='IDENT',
    context_level=0)}
389
390
391 Updated Symbol table global
392 +-----+-----+-----+
-----+-----+
-----+
393 | Name          | Type      | Line Declared   |
Context Level | Value
394 +=====+=====+=====+=====+
=====+=====+
=====+
395 | Monke_beta_0.1 | None     | None:None
|                   | 0 | StatementListNode (type
| ::= 'STATEMENTS', name::= |
396 |                   |           |
|           | 'STATEMENTS', value::= ='deque
| ([]))'           |
397 +-----+-----+-----+
-----+-----+
-----+
398 | a             | None     | 1:4
|                   | 0 | None
399 +-----+-----+-----+
-----+-----+
-----+
400 | b             | None     | 1:6
|                   | 0 | None
401 +-----+-----+-----+

```

```
401
-----
-----+
402
403
404 CHILD deque([IdentifierNode (type::= 'IDENT', name
   ::= 'a', value::= 'None'), IdentifierNode (type::= 'IDENT', name::= 'b', value::= 'None')])
405
406
407 <Type:IDENT, Lexeme:b, BeginPosition:6, LinePosition
   :1> <Type:,, Lexeme:,, BeginPosition:7, LinePosition
   :1>
408
409
410 <Type:,, Lexeme:,, BeginPosition:7, LinePosition:1
   > <Type:IDENT, Lexeme:c, BeginPosition:8,
   LinePosition:1>
411
412
413 IN COMMA
414 <Type:,, Lexeme:,, BeginPosition:7, LinePosition:1
   > <Type:IDENT, Lexeme:c, BeginPosition:8,
   LinePosition:1>
415
416
417 IN COMMA
418 <Type:IDENT, Lexeme:c, BeginPosition:8, LinePosition
   :1> <Type:;, Lexeme:;, BeginPosition:9, LinePosition
   :1>
419
420
421 Dealing with factor: <Type:IDENT, Lexeme:c,
   BeginPosition:8, LinePosition:1>
422 Next dealt terminal/non-term-> token: <Type:;, Lexeme:;, BeginPosition:9, LinePosition:1>
423
424 IdentifierNode (type::= 'IDENT', name::= 'c', value
   :: = 'None')
425
426 <class 'parser.LL1.IdentifierNode'>
```

```

427
428 True
429
430
431 Creating a symbol from Node
432 IdentifierNode (type::= 'IDENT', name::= 'c', value
    :: = 'None')
433 IdentifierNode
434
435
436 Creating a symbol from Node.Value.Token
437 <Type:IDENT, Lexeme:c, BeginPosition:8,
    LinePosition:1>
438
439
440 BEFORE DEFINE _contexts and current_context
441 deque([{'PROGRAM': Symbol(name='Monke_beta_0.1',
    type_='PROGRAM', context_level=0), 'a': Symbol(name
    ='a', type_='IDENT', context_level=0), 'b': Symbol(
    name='b', type_='IDENT', context_level=0)}])
442 {'PROGRAM': Symbol(name='Monke_beta_0.1', type_='
    PROGRAM', context_level=0), 'a': Symbol(name='a',
    type_='IDENT', context_level=0), 'b': Symbol(name='b
    ', type_='IDENT', context_level=0)}
443
444
445 BEFORE DEFINE self: global
446
447 +-----+-----+-----+-----+
        +-----+
        +-----+
448 | Name          | Type      | Line Declared |
    Context Level | Value
        |
449 +=====+=====+=====+=====+
        =====+=====+
        =====+
450 | Monke_beta_0.1 | None     | None:None
            |           0 | StatementListNode (type
            ::= 'STATEMENTS', name::= |
451 |               |           |

```

```

451 | 'STATEMENTS', value:: = 'deque
  ([]))
452 +-----+-----+-----+
-----+
-----+
453 | a | None | 1:4
      |          0 | None
      |
454 +-----+-----+-----+
-----+
-----+
455 | b | None | 1:6
      |          0 | None
      |
456 +-----+-----+-----+
-----+
-----+
457
458
459 SAVING NAME: c
460
461
462 SAVING SYMBOL: Symbol 'c' of type 'IDENT' at context
  level 0
463
464
465 SAVING CONTEXT: {'PROGRAM': Symbol(name='
  Monke_beta_0.1', type_='PROGRAM', context_level=0
), 'a': Symbol(name='a', type_='IDENT',
  context_level=0), 'b': Symbol(name='b', type_='IDENT',
  context_level=0)}
466
467
468 THIS IS CURRENT CONTEXT => {'PROGRAM': Symbol(name='
  Monke_beta_0.1', type_='PROGRAM', context_level=0
), 'a': Symbol(name='a', type_='IDENT',
  context_level=0), 'b': Symbol(name='b', type_='IDENT',
  context_level=0)}
469
470
471 Updated Symbol table global

```

```

472 +-----+-----+-----+
-----+
-----+
473 | Name          | Type     | Line Declared   |
Context Level | Value
               |
474 +=====+=====+=====+=====
=====+=====+=====+=====
=====+
475 | Monke_beta_0.1 | None    | None:None
               |           0 | StatementListNode (type
:::= 'STATEMENTS', name:::= |
476 |               |           |
               |           |
               | 'STATEMENTS', value:: = 'deque
([])')
477 +-----+-----+-----+
-----+
-----+
478 | a            | None    | 1:4
               |           0 | None
               |
479 +-----+-----+-----+
-----+
-----+
480 | b            | None    | 1:6
               |           0 | None
               |
481 +-----+-----+-----+
-----+
-----+
482 | c            | None    | 1:8
               |           0 | None
               |
483 +-----+-----+-----+
-----+
-----+
484
485
486 CHILD deque([IdentifierNode (type::= 'IDENT', name
::= 'a', value::= 'None'), IdentifierNode (type::= 'IDENT',
name::= 'b', value::= 'None'),

```

```
486 IdentifierNode (type::= 'IDENT', name::= 'c', value
    :: ='None'))]
487
488
489 <Type:IDENT, Lexeme:c, BeginPosition:8, LinePosition
    :1> <Type:;, Lexeme:;, BeginPosition:9, LinePosition
    :1>
490
491
492 <Type:;, Lexeme:;, BeginPosition:9, LinePosition:1
    > <Type:IDENT, Lexeme:a, BeginPosition:11,
    LinePosition:1>
493
494
495 Main parser
496 Working from => <Type:IDENT, Lexeme:a, BeginPosition
    :11, LinePosition:1> then <Type:=, Lexeme:=,
    BeginPosition:13, LinePosition:1>
497
498
499 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value:: ='deque([IdentifierNode (type::= 'IDENT', name::= 'a', value:: ='None'),
    IdentifierNode (type::= 'IDENT', name::= 'b', value
    :: ='None'), IdentifierNode (type::= 'IDENT', name
    ::= 'c', value:: ='None')])')])
500
501
502 Parsing <Type:IDENT, Lexeme:a, BeginPosition:11,
    LinePosition:1>
503 Next is <Type:=, Lexeme:=, BeginPosition:13,
    LinePosition:1>
504
505
506 CURRENT TOKEN -> <Type:INT, Lexeme:10, BeginPosition
    :15, LinePosition:1>
507
508 NEXT TOKEN -> <Type:;, Lexeme:;, BeginPosition:17,
    LinePosition:1>
509
510 SHUNTING ON TOP (None, False)
```

```
511
512 CHANGES TO FLAGS (None, False)
513
514 Dealing with factor: <Type:INT, Lexeme:10,
   BeginPosition:15, LinePosition:1>
515 Next dealt terminal/non-term-> token: <Type:;, Lexeme:;, BeginPosition:17, LinePosition:1>
516
517 RETURN BASIC LITERAL NODE <Type:;, Lexeme:;, BeginPosition:17, LinePosition:1> <Type:IDENT, Lexeme:b, BeginPosition:19, LinePosition:1>
518
519
520 This is the expr
521
522 IntegerLiteralNode (type::= 'INT', name::= '10',
   value:: = '10')
523
524 <Type:IDENT, Lexeme:b, BeginPosition:19,
   LinePosition:1>
525
526
527 Main parser
528 Working from => <Type:IDENT, Lexeme:b, BeginPosition :19, LinePosition:1> then <Type:=, Lexeme:=, BeginPosition:21, LinePosition:1>
529
530
531 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value:: ='deque([IdentifierNode (type::= 'IDENT', name::= 'a', value:: ='None'),
   IdentifierNode (type::= 'IDENT', name::= 'b', value :: ='None'), IdentifierNode (type::= 'IDENT', name ::= 'c', value:: ='None')])'), AssignStatementNode (
   type::= 'INT', name::= 'a', left:: ='IdentifierNode (type::= 'IDENT', name::= 'a', value:: ='ExpressionStatementNode (type::= 'INT', name::= '10
   ', value:: ='IntegerLiteralNode (type::= 'INT', name ::= '10', value:: ='10'))')operator:: ='=')right :: ='ExpressionStatementNode (type::= 'INT', name ::= '10', value:: ='IntegerLiteralNode (type::= 'INT
```

```
531 ', name::= '10', value:: ='10'))')value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT', name
    ::= '10', value:: ='10'))])
532
533
534 Parsing <Type:IDENT, Lexeme:b, BeginPosition:19,
    LinePosition:1>
535 Next is <Type:=, Lexeme:=, BeginPosition:21,
    LinePosition:1>
536
537
538 CURRENT TOKEN -> <Type:INT, Lexeme:5, BeginPosition:
    23, LinePosition:1>
539
540 NEXT TOKEN -> <Type:;, Lexeme:;, BeginPosition:24,
    LinePosition:1>
541
542 SHUNTING ON TOP (None, False)
543
544 CHANGES TO FLAGS (None, False)
545
546 Dealing with factor: <Type:INT, Lexeme:5,
    BeginPosition:23, LinePosition:1>
547 Next dealt terminal/non-term-> token: <Type:|,
    Lexeme:;, BeginPosition:24, LinePosition:1>
548
549 RETURN BASIC LITERAL NODE <Type:|,
    Lexeme:|,
    BeginPosition:24, LinePosition:1> <Type:LET, Lexeme:
    let, BeginPosition:26, LinePosition:1>
550
551
552 This is the expr
553
554 IntegerLiteralNode (type::= 'INT', name::= '5',
    value:: ='5')
555
556 <Type:LET, Lexeme:let, BeginPosition:26,
    LinePosition:1>
557
558
```

```

559 Main parser
560 Working from => <Type:LET, Lexeme:let, BeginPosition
:26, LinePosition:1> then <Type:IDENT, Lexeme:d,
BeginPosition:30, LinePosition:1>
561
562
563 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value::= deque([IdentifierNode (type::= 'IDENT', name::= 'a', value::= 'None'),
IdentifierNode (type::= 'IDENT', name::= 'b', value ::= 'None'), IdentifierNode (type::= 'IDENT', name ::= 'c', value::= 'None')])]), AssignStatementNode (
type::= 'INT', name::= 'a', left::= 'IdentifierNode
(type::= 'IDENT', name::= 'a', value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::= 'IntegerLiteralNode (type::= 'INT', name ::= '10', value::= '10')))'operator::= '=')right
::= 'ExpressionStatementNode (type::= 'INT', name ::= '10', value::= 'IntegerLiteralNode (type::= 'INT', name ::= '10', value::= '10')))'value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::= 'IntegerLiteralNode (type::= 'INT', name ::= '10', value::= '10'))), AssignStatementNode (
type::= 'INT', name::= 'b', left::= 'IdentifierNode
(type::= 'IDENT', name::= 'b', value::=
ExpressionStatementNode (type::= 'INT', name::= '5
', value::= 'IntegerLiteralNode (type::= 'INT', name ::= '5', value::= '5')))'operator::= '=')right
::= 'ExpressionStatementNode (type::= 'INT', name ::= '5', value::= 'IntegerLiteralNode (type::= 'INT', name ::= '5', value::= '5')))'value::=
ExpressionStatementNode (type::= 'INT', name::= '5
', value::= 'IntegerLiteralNode (type::= 'INT', name ::= '5', value::= '5'))])
564
565
566 Parsing <Type:LET, Lexeme:let, BeginPosition:26,
LinePosition:1>
567 Next is <Type:IDENT, Lexeme:d, BeginPosition:30,
LinePosition:1>
568

```

```
569
570 Dealing with factor: <Type:IDENT, Lexeme:d,
      BeginPosition:30, LinePosition:1>
571 Next dealt terminal/non-term-> token: <Type:,,,
      Lexeme:,, BeginPosition:31, LinePosition:1>
572
573 IdentifierNode (type::= 'IDENT', name::= 'd', value
      :: ='None')
574
575 <class 'parser.LL1.IdentifierNode'>
576
577 True
578
579
580 Creating a symbol from Node
581 IdentifierNode (type::= 'IDENT', name::= 'd', value
      :: ='None')
582 IdentifierNode
583
584
585 Creating a symbol from Node.Value.Token
586 <Type:IDENT, Lexeme:d, BeginPosition:30,
      LinePosition:1>
587
588
589 BEFORE DEFINE _contexts and current_context
590 deque([{'PROGRAM': Symbol(name='Monke_beta_0.1',
      type_='PROGRAM', context_level=0), 'a': Symbol(name
      ='a', type_='IDENT', context_level=0), 'b': Symbol(
      name='b', type_='IDENT', context_level=0), 'c':
      Symbol(name='c', type_='IDENT', context_level=0)}])
591 {'PROGRAM': Symbol(name='Monke_beta_0.1', type_=
      'PROGRAM', context_level=0), 'a': Symbol(name='a',
      type_='IDENT', context_level=0), 'b': Symbol(name='b
      ', type_='IDENT', context_level=0), 'c': Symbol(name
      ='c', type_='IDENT', context_level=0)}
592
593
594 BEFORE DEFINE self: global
595
596 +-----+-----+-----+-----+
```

596

597	Name	Type	Line Declared	
	Context Level	Value		
598				
599	Monke_beta_0.1	None	None:None	
			0 StatementListNode (type	
		::= 'STATEMENTS', name::=		
600				
		'STATEMENTS', value::= 'deque([
		LetStatementNode (type::=		
601				
		'IDENT', name::= '_a_b_c', value::= '		
		deque([IdentifierNode		
602				
		(type::= 'IDENT', name::= 'a', value		
		::::='None'),		
603				
		IdentifierNode (type::= 'IDENT', name		
		::= 'b', value:::		
604				
		= 'None'), IdentifierNode (type::= 'IDENT		
		' , name::= 'c',		
605				
		value::= 'None'))]),		
		AssignStatementNode (type::= 'INT',		
606				
		name::= 'a', left::= 'IdentifierNode (
		type::= 'IDENT',		
607				
		name::= 'a', value::= '		
		ExpressionStatementNode (type::=		
608				
		'INT', name::= '10', value::= '		
		IntegerLiteralNode (type::=		
609				

```
609 | 'INT', name::= '10', value:: = '10
610 | ')')')operator:: |
611 | | |= '=')right:: ='ExpressionStatementNode (
612 | | type::= 'INT', |
613 | | | name::= '10', value:: =
614 | | | IntegerLiteralNode (type::= 'INT', |
615 | | | | name::= '10', |
616 | | | | | value:: = 'IntegerLiteralNode (type::= 'INT', |
617 | | | | | | name::= '10', |
618 | | | | | | | value:: = '10'))'), AssignStatementNode
619 | | | | | | | (type::= 'INT', |
620 | | | | | | | | name::= 'b', left:: = 'IdentifierNode (
621 | | | | | | | | type::= 'IDENT', |
622 | | | | | | | | | name::= 'b', value:: =
623 | | | | | | | | | ExpressionStatementNode (type::= |
624 | | | | | | | | | | 'INT', name::= '5', value:: =
625 | | | | | | | | | | IntegerLiteralNode (type::= |
626 | | | | | | | | | | | name::= '5', value:: = '5'))'))')
627 | | | | | | | | | operator:: |
628 | | | | | | | | | |= '=')right:: ='ExpressionStatementNode (
629 | | | | | | | | | type::= 'INT', |
630 | | | | | | | | | | name::= '5', value:: =
631 | | | | | | | | | | IntegerLiteralNode (type::= 'INT', |
632 | | | | | | | | | | | name::= '5', value:: = '5'))'))value
633 | | | | | | | | | | | value:: =
```

```
623 |           | = 'ExpressionStatementNode (type::= 'INT'
  ', name::= '5',           |
624 |           |           | value::= 'IntegerLiteralNode (type::= '
  INT', name::= '5',           |
625 |           |           | value::= '5
  ')')')]'))'
626 +-----+-----+
-----+
-----+
627 | a           | None    | 15:1
  |           |          0 |
  ExpressionStatementNode (type::= 'INT', name::= '10
  ',           |
628 |           |           | value::= 'IntegerLiteralNode (type::= '
  INT', name::= '10',           |
629 |           |           | value::= '10
  ')')
630 +-----+-----+
-----+
-----+
631 | b           | None    | 23:1
  |           |          0 |
  ExpressionStatementNode (type::= 'INT', name::= '5
  ', value::=
632 |           |           | = 'IntegerLiteralNode (type::= 'INT',
  name::= '5', value::=
633 |           |           | = '5
  ')')
  |
634 +-----+-----+
-----+
-----+
635 | c           | None    | 1:8
  |           |          0 | None
```

```

635
|
636 +-----+-----+-----+
|-----+
-----+
-----+
637
638
639 SAVING NAME: d
640
641
642 SAVING SYMBOL: Symbol 'd' of type 'IDENT' at context
   level 0
643
644
645 SAVING CONTEXT: {'PROGRAM': Symbol(name='
   Monke_beta_0.1', type_='PROGRAM', context_level=0
   ), 'a': Symbol(name='a', type_='IDENT',
   context_level=0), 'b': Symbol(name='b', type_='IDENT'
   , context_level=0), 'c': Symbol(name='c', type_='
   IDENT', context_level=0)}
646
647
648 THIS IS CURRENT CONTEXT => {'PROGRAM': Symbol(name='
   Monke_beta_0.1', type_='PROGRAM', context_level=0
   ), 'a': Symbol(name='a', type_='IDENT',
   context_level=0), 'b': Symbol(name='b', type_='IDENT'
   , context_level=0), 'c': Symbol(name='c', type_='
   IDENT', context_level=0)}
649
650
651 Updated Symbol table global
652 +-----+-----+-----+
|-----+
-----+
-----+
653 | Name          | Type      | Line Declared    |
| Context Level | Value     |
654 +=====+=====+=====+=====+
|-----+-----+-----+-----+

```

```

654 =====+
655 | Monke_beta_0.1 | None   | None:None
|           | 0 | StatementListNode (type
| ::= 'STATEMENTS', name::=           |
656 |           |   |           |
|           | 'STATEMENTS', value::= 'deque([
| LetStatementNode (type::=           |
657 |           |   |           |
|           | 'IDENT', name::= '_a_b_c', value::= '
| deque([IdentifierNode   |
658 |           |   |           |
|           | (type::= 'IDENT', name::= 'a', value
| ::= 'None'),           |
659 |           |   |           |
|           | IdentifierNode (type::= 'IDENT', name
| ::= 'b', value::=           |
660 |           |   |           |
|           | ='None'), IdentifierNode (type::= 'IDENT
| ', name::= 'c',           |
661 |           |   |           |
|           | value::= 'None')])), |
AssignStatementNode (type::= 'INT',    |
662 |           |   |           |
|           | name::= 'a', left::= 'IdentifierNode (
| type::= 'IDENT',           |
663 |           |   |           |
|           | name::= 'a', value::= '
| ExpressionStatementNode (type::=    |
664 |           |   |           |
|           | 'INT', name::= '10', value::= '
| IntegerLiteralNode (type::=       |
665 |           |   |           |
|           | 'INT', name::= '10', value::= '10
| ')'))'))operator:::           |
666 |           |   |           |
|           | ='=')right::= 'ExpressionStatementNode (
| type::= 'INT',           |
667 |           |   |           |
|           | name::= '10', value::= '
| IntegerLiteralNode (type::= 'INT',   |
668 |           |   |           |

```

```

668           | name::= '10', value:: ='10'))')value
669           ::           |
670           |           |           |
671           |           |           |           |
672           |           |           |           |
673           |           |           |           |
674           |           |           |           |
675           |           |           |           |
676           |           |           |           |
677           |           |           |           |
678           |           |           |           |
679           |           |           |           |
680           |           |           |           |
681           |           |           |           |

```

|

```
682 +-----+-----+-----+
-----+
-----+
683 | a           | None    | 15:1
      |           |          0 |
      ExpressionStatementNode (type::= 'INT', name::= '10
      ',           |
684 |           |           |           |
      | value:: ='IntegerLiteralNode (type::= '
      INT', name::= '10',   |
685 |           |           |           |
      | value:: ='10
      ')')
686 +-----+-----+-----+
-----+
-----+
687 | b           | None    | 23:1
      |           |          0 |
      ExpressionStatementNode (type::= 'INT', name::= '5
      ', value:: |
688 |           |           |           |
      | = 'IntegerLiteralNode (type::= 'INT',
      name::= '5', value:: |
689 |           |           |           |
      | = '5
      ')')
      |
690 +-----+-----+-----+
-----+
-----+
691 | c           | None    | 1:8
      |           |          0 | None
      |
692 +-----+-----+-----+
-----+
-----+
693 | d           | None    | 1:30
      |           |          0 | None
      |
```

```
694 +-----+-----+-----+
-----+
-----+
695
696
697 CHILD deque([IdentifierNode (type::= 'IDENT', name
   ::= 'd', value::= 'None')])
698
699
700 <Type:IDENT, Lexeme:d, BeginPosition:30,
   LinePosition:1> <Type:,, Lexeme:,, BeginPosition:31
   , LinePosition:1>
701
702
703 <Type:,, Lexeme:,, BeginPosition:31, LinePosition:1
   > <Type:IDENT, Lexeme:e, BeginPosition:32,
   LinePosition:1>
704
705
706 IN COMMA
707 <Type:,, Lexeme:,, BeginPosition:31, LinePosition:1
   > <Type:IDENT, Lexeme:e, BeginPosition:32,
   LinePosition:1>
708
709
710 IN COMMA
711 <Type:IDENT, Lexeme:e, BeginPosition:32,
   LinePosition:1> <Type:;, Lexeme:;, BeginPosition:33
   , LinePosition:1>
712
713
714 Dealing with factor: <Type:IDENT, Lexeme:e,
   BeginPosition:32, LinePosition:1>
715 Next dealt terminal/non-term-> token: <Type:|,
   Lexeme:;, BeginPosition:33, LinePosition:1>
716
717 IdentifierNode (type::= 'IDENT', name::= 'e', value
   ::= 'None')
718
719 <class 'parser.LL1.IdentifierNode'>
720
```

```

721 True
722
723
724 Creating a symbol from Node
725 IdentifierNode (type::= 'IDENT', name::= 'e', value
    :: = 'None')
726 IdentifierNode
727
728
729 Creating a symbol from Node.Value.Token
730 <Type:IDENT, Lexeme:e, BeginPosition:32,
    LinePosition:1>
731
732
733 BEFORE DEFINE _contexts and current_context
734 deque([{'PROGRAM': Symbol(name='Monke_beta_0.1',
    type_='PROGRAM', context_level=0), 'a': Symbol(name
    ='a', type_='IDENT', context_level=0), 'b': Symbol(
    name='b', type_='IDENT', context_level=0), 'c':
    Symbol(name='c', type_='IDENT', context_level=0), 'd
    ': Symbol(name='d', type_='IDENT', context_level=0
    )}])
735 {'PROGRAM': Symbol(name='Monke_beta_0.1', type_=
    'PROGRAM', context_level=0), 'a': Symbol(name='a',
    type_='IDENT', context_level=0), 'b': Symbol(name='b
    ', type_='IDENT', context_level=0), 'c': Symbol(name
    ='c', type_='IDENT', context_level=0), 'd': Symbol(
    name='d', type_='IDENT', context_level=0)}
736
737
738 BEFORE DEFINE self: global
739
740 +-----+-----+-----+
    -----+-----+
    -----+
741 | Name          | Type     | Line Declared   |
    Context Level | Value
    |
742 +=====+=====+=====+=====+
    =====+=====+=====+=====+

```

```

742 =====+
743 | Monke_beta_0.1 | None   | None:None
    |                 0 | StatementListNode (type
    ::= 'STATEMENTS', name::=           |
744 |                 |   |           |
    | 'STATEMENTS', value::= 'deque([
    LetStatementNode (type::=      |
745 |                 |   |           |
    | 'IDENT', name::= '_a_b_c', value::= '
    deque([IdentifierNode  |
746 |                 |   |           |
    | (type::= 'IDENT', name::= 'a', value
    ::= 'None'),          |
747 |                 |   |           |
    | IdentifierNode (type::= 'IDENT', name
    ::= 'b', value::=      |
748 |                 |   |           |
    | ='None'), IdentifierNode (type::= 'IDENT
    ', name::= 'c',       |
749 |                 |   |           |
    | value::= 'None')])), |
    AssignStatementNode (type::= 'INT',     |
750 |                 |   |           |
    | name::= 'a', left::= 'IdentifierNode (
    type::= 'IDENT',      |
751 |                 |   |           |
    | name::= 'a', value::= ' |
    ExpressionStatementNode (type::=      |
752 |                 |   |           |
    | 'INT', name::= '10', value::= ' |
    IntegerLiteralNode (type::=      |
753 |                 |   |           |
    | 'INT', name::= '10', value::= '10
    ')'))'))operator::: |
754 |                 |   |           |
    | ='')right::= 'ExpressionStatementNode (
    type::= 'INT',      |
755 |                 |   |           |
    | name::= '10', value::= ' |
    IntegerLiteralNode (type::= 'INT',      |
756 |                 |   |           |

```

```

756           | name::= '10', value:: ='10'))')value
    ::          |
757 |           |           |
    | = 'ExpressionStatementNode (type::= 'INT
    ', name::= '10',      |
758 |           |           |
    | value:: ='IntegerLiteralNode (type::= '
    | INT', name::= '10',      |
759 |           |           |
    | value:: ='10'))'), AssignStatementNode
    (type::= 'INT',      |
760 |           |           |
    | name::= 'b', left:: ='IdentifierNode (
    type::= 'IDENT',      |
761 |           |           |
    | name::= 'b', value:: =
    ExpressionStatementNode (type::=      |
762 |           |           |
    | 'INT', name::= '5', value:: =
    IntegerLiteralNode (type::=      |
763 |           |           |
    | 'INT', name::= '5', value:: ='5'))')
operator:::
764 |           |           |
    | ='')right:: ='ExpressionStatementNode (
    type::= 'INT',      |
765 |           |           |
    | name::= '5', value:: =
    IntegerLiteralNode (type::= 'INT',      |
766 |           |           |
    | name::= '5', value:: ='5'))')value
    ::          |
767 |           |           |
    | = 'ExpressionStatementNode (type::= 'INT
    ', name::= '5',      |
768 |           |           |
    | value:: ='IntegerLiteralNode (type::= '
    | INT', name::= '5',      |
769 |           |           |
    | value:: ='5
    ')'))])')

```

```
770 +-----+-----+-----+
-----+
-----+
771 | a           | None    | 15:1
      |           |          0 |
      |
      | ExpressionStatementNode (type::= 'INT', name::= '10
      ',           |
572 |           |           |           |
      | value:: ='IntegerLiteralNode (type::= '
      INT', name::= '10',   |
573 |           |           |           |
      | value:: ='10
      ')')
574 +-----+-----+-----+
-----+
-----+
575 | b           | None    | 23:1
      |           |          0 |
      |
      | ExpressionStatementNode (type::= 'INT', name::= '5
      ', value:: |
576 |           |           |           |
      |           | ='IntegerLiteralNode (type::= 'INT',
      name::= '5', value:: |
577 |           |           |           |
      |           | = '5
      ')')
      |
578 +-----+-----+-----+
-----+
-----+
579 | c           | None    | 1:8
      |           |          0 | None
      |
580 +-----+-----+-----+
-----+
-----+
581 | d           | None    | 1:30
      |           |          0 | None
      |
```

```

782 +-----+-----+-----+
-----+
-----+
783
784
785 SAVING NAME: e
786
787
788 SAVING SYMBOL: Symbol 'e' of type 'IDENT' at context
    level 0
789
790
791 SAVING CONTEXT: {'PROGRAM': Symbol(name='
    Monke_beta_0.1', type_='PROGRAM', context_level=0
), 'a': Symbol(name='a', type_='IDENT',
    context_level=0), 'b': Symbol(name='b', type_='IDENT',
    context_level=0), 'c': Symbol(name='c', type_='
    IDENT', context_level=0), 'd': Symbol(name='d',
    type_='IDENT', context_level=0)}
792
793
794 THIS IS CURRENT CONTEXT => {'PROGRAM': Symbol(name='
    Monke_beta_0.1', type_='PROGRAM', context_level=0
), 'a': Symbol(name='a', type_='IDENT',
    context_level=0), 'b': Symbol(name='b', type_='IDENT',
    context_level=0), 'c': Symbol(name='c', type_='
    IDENT', context_level=0), 'd': Symbol(name='d',
    type_='IDENT', context_level=0)}
795
796
797 Updated Symbol table global
798 +-----+-----+-----+
-----+
-----+
799 | Name          | Type      | Line Declared   |
| Context Level | Value     |
|
800 +=====+=====+=====+=====+
=====+=====+=====+=====+
=====+=====+

```

```

801 | Monke_beta_0.1 | None    | None:None
     |                 0 | StatementListNode (type
     ::= 'STATEMENTS', name::=
802 |                 |         |
     | 'STATEMENTS', value:: = 'deque([
     LetStatementNode (type::=      |
803 |                 |         |
     | 'IDENT', name::= '_a_b_c', value:: = '
     deque([IdentifierNode   |
804 |                 |         |
     | (type::= 'IDENT', name::= 'a', value
     :: = 'None'),           |
805 |                 |         |
     | IdentifierNode (type::= 'IDENT', name
     ::= 'b', value::   |
806 |                 |         |
     | ='None'), IdentifierNode (type::= 'IDENT
     ', name::= 'c',       |
807 |                 |         |
     | value:: = 'None')])'),
     AssignStatementNode (type::= 'INT',      |
808 |                 |         |
     | name::= 'a', left:: ='IdentifierNode (
     type::= 'IDENT',       |
809 |                 |         |
     | name::= 'a', value:: =
     ExpressionStatementNode (type::=      |
810 |                 |         |
     | 'INT', name::= '10', value:: =
     IntegerLiteralNode (type::=      |
811 |                 |         |
     | 'INT', name::= '10', value:: = '10
     ')'))')operator:: |
812 |                 |         |
     | ='=')right:: ='ExpressionStatementNode (
     type::= 'INT',       |
813 |                 |         |
     | name::= '10', value:: =
     IntegerLiteralNode (type::= 'INT',      |
814 |                 |         |
     | name::= '10', value:: = '10')'))value

```

```
814 ::          |
815 |          |          |
816 |          |          |          |
817 |          |          |          |          |
818 |          |          |          |          |
819 |          |          |          |          |
820 |          |          |          |          |
821 |          |          |          |          |
822 |          |          |          |          |
823 |          |          |          |          |
824 |          |          |          |          |
825 |          |          |          |          |
826 |          |          |          |          |
827 |          |          |          |          |
828 +-----+-----+-----+
```

```
828 +-----+
829 | a           | None    | 15:1
     |           |          0 |
     ExpressionStatementNode (type::= 'INT', name::= '10
     ',           |
830 |           |           |           |
     | value::= 'IntegerLiteralNode (type::= 'INT',
     INT', name::= '10',   |
831 |           |           |           |
     | value::= '10
     ')')
832 +-----+-----+
833 | b           | None    | 23:1
     |           |          0 |
     ExpressionStatementNode (type::= 'INT', name::= '5
     ', value:: |
834 |           |           |           |
     | = 'IntegerLiteralNode (type::= 'INT',
     name::= '5', value:: |
835 |           |           |           |
     | = '5
     ')')
     |
836 +-----+-----+
837 | c           | None    | 1:8
     |           |          0 | None
     |
838 +-----+-----+
839 | d           | None    | 1:30
     |           |          0 | None
```

```

840 +-----+-----+-----+
-----+
-----+
841 | e | None | 1:32
      |          0 | None
      |
842 +-----+-----+-----+
-----+
-----+
843
844
845 CHILD deque([IdentifierNode (type::= 'IDENT', name
     ::= 'd', value:: = 'None'), IdentifierNode (type::= 'IDENT', name::= 'e', value:: = 'None')])
846
847
848 <Type:IDENT, Lexeme:e, BeginPosition:32,
     LinePosition:1> <Type:;, Lexeme:;, BeginPosition:33
     , LinePosition:1>
849
850
851 <Type:;, Lexeme:;, BeginPosition:33, LinePosition:1
     > <Type:IDENT, Lexeme:c, BeginPosition:35,
     LinePosition:1>
852
853
854 Main parser
855 Working from => <Type:IDENT, Lexeme:c, BeginPosition
     :35, LinePosition:1> then <Type:=, Lexeme:=,
     BeginPosition:36, LinePosition:1>
856
857
858 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value:: = 'deque([IdentifierNode (type::= 'IDENT', name::= 'a', value:: = 'None'),
     IdentifierNode (type::= 'IDENT', name::= 'b', value
     :: = 'None'), IdentifierNode (type::= 'IDENT', name
     ::= 'c', value:: = 'None')])'), AssignStatementNode (
     type::= 'INT', name::= 'a', left:: = 'IdentifierNode
     (type::= 'IDENT', name::= 'a', value:: ='
```

```

858 ExpressionStatementNode (type::= 'INT', name::= '10
  ', value::='IntegerLiteralNode (type::= 'INT', name
  ::= '10', value::='10'))operator::='=')right
  ::= 'ExpressionStatementNode (type::= 'INT', name
  ::= '10', value::='IntegerLiteralNode (type::= 'INT',
  name::= '10', value::='10'))')value::=
  ExpressionStatementNode (type::= 'INT', name::= '10
  ', value::='IntegerLiteralNode (type::= 'INT', name
  ::= '10', value::='10'))'), AssignStatementNode (
  type::= 'INT', name::= 'b', left::='IdentifierNode
  (type::= 'IDENT', name::= 'b', value::='
  ExpressionStatementNode (type::= 'INT', name::= '5
  ', value::='IntegerLiteralNode (type::= 'INT', name
  ::= '5', value::='5'))')operator::='=')right
  ::= 'ExpressionStatementNode (type::= 'INT', name
  ::= '5', value::='IntegerLiteralNode (type::= 'INT',
  name::= '5', value::='5'))')value::=
  ExpressionStatementNode (type::= 'INT', name::= '5
  ', value::='IntegerLiteralNode (type::= 'INT', name
  ::= '5', value::='5'))'), LetStatementNode (type
  ::= 'IDENT', name::= '_d_e', value::='deque([
  IdentifierNode (type::= 'IDENT', name::= 'd', value
  ::='None'), IdentifierNode (type::= 'IDENT', name
  ::= 'e', value::='None')]))])
859
860
861 Parsing <Type:IDENT, Lexeme:c, BeginPosition:35,
  LinePosition:1>
862 Next is <Type:=, Lexeme:=, BeginPosition:36,
  LinePosition:1>
863
864
865 CURRENT TOKEN -> <Type:IDENT, Lexeme:a,
  BeginPosition:37, LinePosition:1>
866
867 NEXT TOKEN -> <Type:+, Lexeme:+, BeginPosition:38,
  LinePosition:1>
868
869 SHUNTING ON TOP (None, False)
870
871 CHANGES TO FLAGS (True, False)

```

```
872
873 Dealing with factor: <Type:IDENT, Lexeme:a,
   BeginPosition:37, LinePosition:1>
874 Next dealt terminal/non-term-> token: <Type:+,
   Lexeme:+, BeginPosition:38, LinePosition:1>
875
876 **MURKY**
877 AssignStatementNode (type::= 'INT', name::= 'a',
   left::= 'IdentifierNode (type::= 'IDENT', name::= 'a
   ', value::= 'ExpressionStatementNode (type::= 'INT
   ', name::= '10', value::= 'IntegerLiteralNode (type
   ::= 'INT', name::= '10', value::= '10'))')
   operator::= '=')right::= 'ExpressionStatementNode (
   type::= 'INT', name::= '10', value::=
   IntegerLiteralNode (type::= 'INT', name::= '10',
   value::= '10'))value::= 'ExpressionStatementNode
   (type::= 'INT', name::= '10', value::=
   IntegerLiteralNode (type::= 'INT', name::= '10',
   value::= '10'))'
878 <Type:IDENT, Lexeme:a, BeginPosition:37,
   LinePosition:1>
879 <Type:IDENT, Lexeme:a, BeginPosition:37,
   LinePosition:1>
880 <Type:+, Lexeme:+, BeginPosition:38, LinePosition:1>
881 **
882
883
884 CURRENT
885 <Type:IDENT, Lexeme:a, BeginPosition:37,
   LinePosition:1>
886 NEXT
887 <Type:+, Lexeme:+, BeginPosition:38, LinePosition:1>
888
889 CURRENT
890 <Type:+, Lexeme:+, BeginPosition:38, LinePosition:1>
891 NEXT
892 <Type:INT, Lexeme:4, BeginPosition:39, LinePosition:
   1>
893
894 CURRENT
895 <Type:INT, Lexeme:4, BeginPosition:39, LinePosition:
```

```
895 1>
896 NEXT
897 <Type:+, Lexeme:+, BeginPosition:40, LinePosition:1>
898
899 CURRENT
900 <Type:+, Lexeme:+, BeginPosition:40, LinePosition:1>
901 NEXT
902 <Type:IDENT, Lexeme:b, BeginPosition:41,
LinePosition:1>
903
904 CURRENT
905 <Type:IDENT, Lexeme:b, BeginPosition:41,
LinePosition:1>
906 NEXT
907 <Type:;, Lexeme:;, BeginPosition:42, LinePosition:1>
908
909 Recovered from error: Unexpected token <Type:IDENT,
Lexeme:a, BeginPosition:37, LinePosition:1> with
lexeme 'a' at position 37 on line 1.
910
911
912 DONE ALL EXPRESSION ATOMS
913 OPERATOR STACK: deque([])
914
915 OUTPUT STACK: deque([AssignStatementNode (type::= 'INT',
name::= 'a', left:: =IdentifierNode (type
::= 'IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT', name
::= '10', value:: ='10'))')operator:: ='=')right
:: =ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT', name
::= '10', value:: ='10'))])
916
917
918 CLEAR OPS STACK
919 OPERATOR STACK deque([])
920
```

```

921
922 OUTPUT STACK deque([AssignStatementNode (type::= 'INT', name::= 'a', left::= 'IdentifierNode (type::= 'IDENT', name::= 'a', value::= 'ExpressionStatementNode (type::= 'INT', name::= '10', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10', value::= '10'))')operator::= '=')right::= 'ExpressionStatementNode (type::= 'INT', name::= '10', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10', value::= '10'))')value::= 'ExpressionStatementNode (type::= 'INT', name::= '10', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10', value::= '10'))])
923
924
925 This is the expr
926
927 AssignStatementNode (type::= 'INT', name::= 'a', left::= 'IdentifierNode (type::= 'IDENT', name::= 'a', value::= 'ExpressionStatementNode (type::= 'INT', name::= '10', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10', value::= '10'))')operator::= '=')right::= 'ExpressionStatementNode (type::= 'INT', name::= '10', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10', value::= '10'))')value::= 'ExpressionStatementNode (type::= 'INT', name::= '10', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10', value::= '10'))')
928
929 <Type:;, Lexeme:;, BeginPosition:43, LinePosition:1>
930
931
932 Main parser
933 Working from => <Type:IDENT, Lexeme:d, BeginPosition:46, LinePosition:1> then <Type:=, Lexeme:=, BeginPosition:48, LinePosition:1>
934
935
936 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value::= 'deque([IdentifierNode (type::= '

```

```

936 IDENT', name::= 'a', value::= 'None'),
IdentifierNode (type::= 'IDENT', name::= 'b', value
::= 'None'), IdentifierNode (type::= 'IDENT', name
::= 'c', value::= 'None')]])), AssignStatementNode (
type::= 'INT', name::= 'a', left::= 'IdentifierNode
(type::= 'IDENT', name::= 'a', value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::= 'IntegerLiteralNode (type::= 'INT', name
::= '10', value::= '10')))'operator::= '=')right
::= 'ExpressionStatementNode (type::= 'INT', name
::= '10', value::= 'IntegerLiteralNode (type::= 'INT
', name::= '10', value::= '10')))'value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::= 'IntegerLiteralNode (type::= 'INT', name
::= '10', value::= '10'))', AssignStatementNode (
type::= 'INT', name::= 'b', left::= 'IdentifierNode
(type::= 'IDENT', name::= 'b', value::=
ExpressionStatementNode (type::= 'INT', name::= '5
', value::= 'IntegerLiteralNode (type::= 'INT', name
::= '5', value::= '5')))'operator::= '=')right
::= 'ExpressionStatementNode (type::= 'INT', name
::= '5', value::= 'IntegerLiteralNode (type::= 'INT
', name::= '5', value::= '5')))'value::=
ExpressionStatementNode (type::= 'INT', name::= '5
', value::= 'IntegerLiteralNode (type::= 'INT', name
::= '5', value::= '5'))', LetStatementNode (type
::= 'IDENT', name::= '_d_e', value::= 'deque([
IdentifierNode (type::= 'IDENT', name::= 'd', value
::= 'None'), IdentifierNode (type::= 'IDENT', name
::= 'e', value::= 'None')]))', AssignStatementNode (
type::= 'INT', name::= 'c', left::= 'IdentifierNode
(type::= 'IDENT', name::= 'c', value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::= 'AssignStatementNode (type::= 'INT',
name::= 'a', left::= 'IdentifierNode (type::= 'IDENT
', name::= 'a', value::= 'ExpressionStatementNode (
type::= 'INT', name::= '10', value::=
IntegerLiteralNode (type::= 'INT', name::= '10',
value::= '10')))'operator::= '=')right::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::= 'IntegerLiteralNode (type::= 'INT', name

```

```

936 ::= '10', value:: ='10'))')value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT', name
    ::= '10', value:: ='10'))')')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='AssignStatementNode (type
::= 'INT', name::= 'a', left:: ='IdentifierNode (
type::= 'IDENT', name::= 'a', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT', name
    ::= '10', value:: ='10'))')operator:: ='=')right
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= 'INT
    ', name::= '10', value:: ='10'))')value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::= 'IDENT
    ', name::= 'a', value:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')operator:: ='=')right:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT', name
    ::= '10', value:: ='10'))')value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT', name
    ::= '10', value:: ='10'))')]))

937
938
939 Parsing <Type:IDENT, Lexeme:d, BeginPosition:46,
LinePosition:1>
940 Next is <Type:=, Lexeme:=, BeginPosition:48,
LinePosition:1>
941
942
943 CURRENT TOKEN -> <Type:IDENT, Lexeme:c,
BeginPosition:50, LinePosition:1>
944

```

```
945 NEXT TOKEN -> <Type:+, Lexeme:+, BeginPosition:52,
LinePosition:1>
946
947 SHUNTING ON TOP (False, False)
948
949 CHANGES TO FLAGS (True, False)
950
951 Dealing with factor: <Type:IDENT, Lexeme:c,
BeginPosition:50, LinePosition:1>
952 Next dealt terminal/non-term-> token: <Type:+,
Lexeme:+, BeginPosition:52, LinePosition:1>
953
954 **MURKY**
955 AssignStatementNode (type::= 'INT', name::= 'c',
left::='IdentifierNode (type::= 'IDENT', name::= 'c
', value::='ExpressionStatementNode (type::= 'INT
', name::= '10', value::='AssignStatementNode (type
::= 'INT', name::= 'a', left::='IdentifierNode (
type::= 'IDENT', name::= 'a', value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::='IntegerLiteralNode (type::= 'INT', name
::= '10', value::='10'))')operator::='=')right
::='ExpressionStatementNode (type::= 'INT', name
::= '10', value::='IntegerLiteralNode (type::= 'INT
', name::= '10', value::='10'))')value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::='IntegerLiteralNode (type::= 'INT', name
::= '10', value::='10'))')operator::='=')
right::='ExpressionStatementNode (type::= 'INT',
name::= '10', value::='AssignStatementNode (type
::= 'INT', name::= 'a', left::='IdentifierNode (
type::= 'IDENT', name::= 'a', value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::='IntegerLiteralNode (type::= 'INT', name
::= '10', value::='10'))')operator::='=')right
::='ExpressionStatementNode (type::= 'INT', name
::= '10', value::='IntegerLiteralNode (type::= 'INT
', name::= '10', value::='10'))')value::=
ExpressionStatementNode (type::= 'INT', name::= '10
', value::='IntegerLiteralNode (type::= 'INT', name
::= '10', value::='10'))')value::='
```

```
955 ExpressionStatementNode (type::= 'INT', name::= '10
  ', value::='AssignStatementNode (type::= 'INT',
  name::= 'a', left::='IdentifierNode (type::= 'IDENT
  ', name::= 'a', value::='ExpressionStatementNode (
  type::= 'INT', name::= '10', value::=
  IntegerLiteralNode (type::= 'INT', name::= '10',
  value::='10'))'))operator::='=')right::='
  ExpressionStatementNode (type::= 'INT', name::= '10
  ', value::='IntegerLiteralNode (type::= 'INT', name
  ::= '10', value::='10'))value::='
  ExpressionStatementNode (type::= 'INT', name::= '10
  ', value::='IntegerLiteralNode (type::= 'INT', name
  ::= '10', value::='10'))')
956 <Type:IDENT, Lexeme:c, BeginPosition:50,
  LinePosition:1>
957 <Type:IDENT, Lexeme:c, BeginPosition:50,
  LinePosition:1>
958 <Type:+, Lexeme:+, BeginPosition:52, LinePosition:1>
959 **
960
961
962 CURRENT
963 <Type:IDENT, Lexeme:c, BeginPosition:50,
  LinePosition:1>
964 NEXT
965 <Type:+, Lexeme:+, BeginPosition:52, LinePosition:1>
966
967 CURRENT
968 <Type:+, Lexeme:+, BeginPosition:52, LinePosition:1>
969 NEXT
970 <Type:INT, Lexeme:10, BeginPosition:54, LinePosition
  :1>
971
972 CURRENT
973 <Type:INT, Lexeme:10, BeginPosition:54, LinePosition
  :1>
974 NEXT
975 <Type:;, Lexeme:;, BeginPosition:56, LinePosition:1>
976
977 Recovered from error: Unexpected token <Type:IDENT,
  Lexeme:c, BeginPosition:50, LinePosition:1> with
```

```
977 lexeme 'c' at position 50 on line 1.  
978  
979  
980 DONE ALL EXPRESSION ATOMS  
981 OPERATOR STACK: deque([])  
982  
983 OUTPUT STACK: deque([AssignStatementNode (type::= 'INT', name::= 'c', left::= 'IdentifierNode (type ::= 'IDENT', name::= 'c', value::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'AssignStatementNode (type::= 'INT', name::= 'a', left::= 'IdentifierNode (type::= 'IDENT', name::= 'a', value::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10 ', value::= '10'))')operator::= '=' right::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10 ', value::= '10'))value ::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10 ', value::= '10'))')operator::= '=' right::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'AssignStatementNode (type::= 'INT', name::= 'a', left::= 'IdentifierNode (type::= 'IDENT', name::= 'a', value::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10 ', value::= '10'))')operator::= '=' right::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10 ', value::= '10'))value ::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10 ', value::= '10'))')value::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'IntegerLiteralNode (type::= 'INT', name::= '10 ', value::= '10'))')value::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'AssignStatementNode (type::= 'INT', name::= 'a', left::= 'IdentifierNode (type::= 'IDENT', name::= 'a', value::= 'ExpressionStatementNode (type::= 'INT', name::= '10 ', value::= 'AssignStatementNode (type::= 'INT', name::= 'a', left::= 'IdentifierNode (type::= 'IDENT', name::= 'a', value::= 'ExpressionStatementNode (type::= 'INT', name::= '10 '))')')')')
```

```
983 ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= '
    INT', name::= '10', value:: ='10'))'))'))])
984
985
986 CLEAR OPS STACK
987 OPERATOR STACK deque([])
988
989
990 OUTPUT STACK deque([AssignStatementNode (type::= '
    INT', name::= 'c', left:: ='IdentifierNode (type
    ::= 'IDENT', name::= 'c', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='AssignStatementNode (type::= 'INT',
    name::= 'a', left:: ='IdentifierNode (type::= '
        IDENT', name::= 'a', value:: =
        ExpressionStatementNode (type::= 'INT', name::= '10
        ', value:: ='IntegerLiteralNode (type::= 'INT',
        name::= '10', value:: ='10'))')operator:: ='=')
    right:: ='ExpressionStatementNode (type::= 'INT',
        name::= '10', value:: ='IntegerLiteralNode (type
        ::= 'INT', name::= '10', value:: ='10'))value
        :: ='ExpressionStatementNode (type::= 'INT', name
        ::= '10', value:: ='IntegerLiteralNode (type::= '
            INT', name::= '10', value:: ='10'))'))')
    operator:: ='=')right:: ='ExpressionStatementNode (
        type::= 'INT', name::= '10', value:: =
        AssignStatementNode (type::= 'INT', name::= 'a',
        left:: ='IdentifierNode (type::= 'IDENT', name::= '
            a', value:: =
            ExpressionStatementNode (type::= 'INT
            ', name::= '10', value:: ='IntegerLiteralNode (type
            ::= 'INT', name::= '10', value:: ='10'))')
        operator:: ='=')right:: ='ExpressionStatementNode (
            type::= 'INT', name::= '10', value:: =
            IntegerLiteralNode (type::= 'INT', name::= '10',
            value:: ='10'))value:: ='ExpressionStatementNode
```

```

990 (type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))'))value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='AssignStatementNode (type::= 'INT',
    name::= 'a', left:: ='IdentifierNode (type::=
    IDENT', name::= 'a', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')operator:: ='=')
    right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type::=
    INT', name::= '10', value:: ='10'))'))])
991
992
993 This is the expr
994
995 AssignStatementNode (type::= 'INT', name::= 'c',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'c',
    value:: ='ExpressionStatementNode (type::= 'INT
    ', name::= '10', value:: ='AssignStatementNode (
    type::= 'INT', name::= 'a', left:: ='IdentifierNode
    (type::= 'IDENT', name::= 'a', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')')operator:: ='=')
    right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type::
    = 'INT', name::= '10', value:: ='10'))')value
    :: ='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::=
    INT', name::= '10', value:: ='10'))'))')
    operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    AssignStatementNode (type::= 'INT', name::= 'a',
    left:: ='IdentifierNode (type::= 'IDENT', name::=
    'a', value:: ='ExpressionStatementNode (type::=
    INT', name::= '10', value:: ='IntegerLiteralNode (type
    
```

```

995 ::= 'INT', name::= '10', value:: = '10'))')))
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: ='ExpressionStatementNode
(type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::=
'IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type::=
'INT', name::= '10', value:: ='10'))')value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::=
'INT', name::= '10', value:: ='10'))')')
996
997 <Type:IDENT, Lexeme:e, BeginPosition:57,
LinePosition:1>
998
999
1000 Main parser
1001 Working from => <Type:IDENT, Lexeme:e,
BeginPosition:57, LinePosition:1> then <Type:=,
Lexeme:=, BeginPosition:59, LinePosition:1>
1002
1003
1004 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value:: ='deque([IdentifierNode (type::= 'IDENT', name::= 'a', value:: ='None'),
IdentifierNode (type::= 'IDENT', name::= 'b', value :: ='None'), IdentifierNode (type::= 'IDENT', name ::= 'c', value:: ='None')])']), AssignStatementNode
(type::= 'INT', name::= 'a', left:: =
IdentifierNode (type::= 'IDENT', name::= 'a', value :: ='ExpressionStatementNode (type::= 'INT', name

```

```

1004 ::= '10', value::='IntegerLiteralNode (type::= 'INT', name::= '10', value::='10'))'operator
::='=')right::='ExpressionStatementNode (type
::= 'INT', name::= '10', value::='
IntegerLiteralNode (type::= 'INT', name::= '10',
value::='10'))'value::='ExpressionStatementNode
(type::= 'INT', name::= '10', value::='
IntegerLiteralNode (type::= 'INT', name::= '10',
value::='10'))'), AssignStatementNode (type::= 'INT',
name::= 'b', left::='IdentifierNode (type
::= 'IDENT', name::= 'b', value::='
ExpressionStatementNode (type::= 'INT', name::= '5',
value::='IntegerLiteralNode (type::= 'INT',
name::= '5', value::='5'))'operator::='=')
right::='ExpressionStatementNode (type::= 'INT',
name::= '5', value::='IntegerLiteralNode (type
::= 'INT', name::= '5', value::='5'))value::='
ExpressionStatementNode (type::= 'INT', name::= '5',
value::='IntegerLiteralNode (type::= 'INT',
name::= '5', value::='5'))'), LetStatementNode (
type::= 'IDENT', name::= '_d_e', value::='deque([
IdentifierNode (type::= 'IDENT', name::= 'd', value
::='None'), IdentifierNode (type::= 'IDENT', name
::= 'e', value::='None')])'), AssignStatementNode
(type::= 'INT', name::= 'c', left::='
IdentifierNode (type::= 'IDENT', name::= 'c', value
::='ExpressionStatementNode (type::= 'INT', name
::= '10', value::='AssignStatementNode (type::= 'INT',
name::= 'a', left::='IdentifierNode (type
::= 'IDENT', name::= 'a', value::='
ExpressionStatementNode (type::= 'INT', name::= '10',
value::='IntegerLiteralNode (type::= 'INT',
name::= '10', value::='10'))'operator::='=')
right::='ExpressionStatementNode (type::= 'INT',
name::= '10', value::='IntegerLiteralNode (type::= 'INT',
name::= '10', value::='10'))value
::='ExpressionStatementNode (type::= 'INT', name
::= '10', value::='IntegerLiteralNode (type::= 'INT',
name::= '10', value::='10'))'))')
operator::='=')right::='ExpressionStatementNode (
type::= 'INT', name::= '10', value::='

```

```
1004 AssignStatementNode (type::= 'INT', name::= 'a',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
        value:: ='ExpressionStatementNode (type::= 'INT',
            name::= '10', value:: ='IntegerLiteralNode (type
                ::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
        value:: ='10'))value:: ='ExpressionStatementNode
    (type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
        value:: ='10'))')value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10',
        value:: ='AssignStatementNode (type::= 'INT',
            name::= 'a', left:: ='IdentifierNode (type::= 'IDENT',
                name::= 'a', value:: =
                ExpressionStatementNode (type::= 'INT', name::= '10',
                    value:: ='IntegerLiteralNode (type::= 'INT',
                        name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
        ::= 'INT', name::= '10', value:: ='10'))value
    :: ='ExpressionStatementNode (type::= 'INT', name
        ::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
            name::= '10', value:: ='10'))'), AssignStatementNode (type::= 'INT',
    name::= 'd',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'd',
        value:: ='ExpressionStatementNode (type::= 'INT',
            name::= '10', value:: ='AssignStatementNode (
                type::= 'INT', name::= 'c', left:: ='IdentifierNode
                    (type::= 'IDENT', name::= 'c', value:: =
                    ExpressionStatementNode (type::= 'INT', name::= '10',
                        value:: ='AssignStatementNode (type::= 'INT',
                            name::= 'a', left:: ='IdentifierNode (type::= 'IDENT',
                                name::= 'a', value:: =
                                ExpressionStatementNode (type::= 'INT', name::= '10',
                                    value:: ='IntegerLiteralNode (type::= 'INT',
                                        name::= '10', value:: ='10'))')operator:: ='=')
                            right:: ='ExpressionStatementNode (type::= 'INT',
                                name::= '10', value:: ='IntegerLiteralNode (type
                                    ::= 'INT', name::= '10', value:: ='10'))')value
```

```
1004 :: ='ExpressionStatementNode (type::= 'INT', name
     ::= '10', value:: ='IntegerLiteralNode (type::= '
     INT', name::= '10', value:: ='10'))'))'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'a',
left:: ='IdentifierNode (type::= 'IDENT', name::= '
a', value:: ='ExpressionStatementNode (type::= 'INT
', name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: ='ExpressionStatementNode
(type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::= '
IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= '
INT', name::= '10', value:: ='10'))')')')')')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'c',
left:: ='IdentifierNode (type::= 'IDENT', name::= '
c', value:: ='ExpressionStatementNode (type::= 'INT
', name::= '10', value:: ='AssignStatementNode (
type::= 'INT', name::= 'a', left:: ='IdentifierNode
(type::= 'IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
```

```
1004 right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))value
    :: ='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))'))'))'
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    AssignStatementNode (type::= 'INT', name::= 'a',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
    value:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))'))'
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))')value:: ='ExpressionStatementNode
    (type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))')value:: ='ExpressionStatementNode
    (type::= 'INT', name::= '10', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10',
    value:: ='AssignStatementNode (type::= 'INT',
    name::= 'a', left:: ='IdentifierNode (type::= 'IDENT',
    name::= 'a', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10',
    value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))'))operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))')value
    :: ='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))'))'))')value
    :: ='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='AssignStatementNode (type::= 'INT',
    name::= 'c', left:: ='IdentifierNode (type
    ::= 'IDENT', name::= 'c', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10',
    value:: ='AssignStatementNode (type::= 'INT',
    name::= 'a', left:: ='IdentifierNode (type::= 'IDENT',
    name::= 'a', value:: ='
```

```

1004 ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))')value
:: ='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::= '
    INT', name::= '10', value:: ='10'))')')')
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'a',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
    value:: ='ExpressionStatementNode (type::= 'INT
        ', name::= '10', value:: ='IntegerLiteralNode (type
        ::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))')value:: ='ExpressionStatementNode
    (type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))')')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='AssignStatementNode (type::= 'INT',
    name::= 'a', left:: ='IdentifierNode (type::= '
    IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))')value
:: ='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::= '
    INT', name::= '10', value:: ='10'))')')')'])
1005
1006
1007 Parsing <Type:IDENT, Lexeme:e, BeginPosition:57,
    LinePosition:1>
1008 Next is <Type:=, Lexeme:=, BeginPosition:59,

```

```
1008 LinePosition:1>
1009
1010
1011 CURRENT TOKEN -> <Type:IDENT, Lexeme:a,
    BeginPosition:61, LinePosition:1>
1012
1013 NEXT TOKEN -> <Type:+, Lexeme:+, BeginPosition:63,
    LinePosition:1>
1014
1015 SHUNTING ON TOP (False, False)
1016
1017 CHANGES TO FLAGS (True, False)
1018
1019 Dealing with factor: <Type:IDENT, Lexeme:a,
    BeginPosition:61, LinePosition:1>
1020 Next dealt terminal/non-term-> token: <Type:+,
    Lexeme:+, BeginPosition:63, LinePosition:1>
1021
1022 **MURKY**
1023 AssignStatementNode (type::= 'INT', name::= 'a',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
        value:: ='ExpressionStatementNode (type::= 'INT
        ', name::= '10', value:: ='IntegerLiteralNode (type
        ::= 'INT', name::= '10', value:: ='10'))')
    operator:: ='=')right:: ='ExpressionStatementNode (
        type::= 'INT', name::= '10', value:: =
        IntegerLiteralNode (type::= 'INT', name::= '10',
            value:: ='10'))value:: ='ExpressionStatementNode
        (type::= 'INT', name::= '10', value:: =
        IntegerLiteralNode (type::= 'INT', name::= '10',
            value:: ='10'))'
1024 <Type:IDENT, Lexeme:a, BeginPosition:61,
    LinePosition:1>
1025 <Type:IDENT, Lexeme:a, BeginPosition:61,
    LinePosition:1>
1026 <Type:+, Lexeme:+, BeginPosition:63, LinePosition:1
    >
1027 **
1028
1029
1030 CURRENT
```

```
1031 <Type:IDENT, Lexeme:a, BeginPosition:61,
      LinePosition:1>
1032 NEXT
1033 <Type:+, Lexeme:+, BeginPosition:63, LinePosition:1
      >
1034
1035 CURRENT
1036 <Type:+, Lexeme:+, BeginPosition:63, LinePosition:1
      >
1037 NEXT
1038 <Type:IDENT, Lexeme:d, BeginPosition:64,
      LinePosition:1>
1039
1040 CURRENT
1041 <Type:IDENT, Lexeme:d, BeginPosition:64,
      LinePosition:1>
1042 NEXT
1043 <Type:*, Lexeme:*, BeginPosition:65, LinePosition:1
      >
1044
1045 CURRENT
1046 <Type:*, Lexeme:*, BeginPosition:65, LinePosition:1
      >
1047 NEXT
1048 <Type:IDENT, Lexeme:b, BeginPosition:66,
      LinePosition:1>
1049
1050 CURRENT
1051 <Type:IDENT, Lexeme:b, BeginPosition:66,
      LinePosition:1>
1052 NEXT
1053 <Type:/, Lexeme:/, BeginPosition:67, LinePosition:1
      >
1054
1055 CURRENT
1056 <Type:/, Lexeme:/, BeginPosition:67, LinePosition:1
      >
1057 NEXT
1058 <Type:IDENT, Lexeme:c, BeginPosition:68,
      LinePosition:1>
1059
```

```
1060 CURRENT
1061 <Type:IDENT, Lexeme:c, BeginPosition:68,
      LinePosition:1>
1062 NEXT
1063 <Type:;, Lexeme:;, BeginPosition:69, LinePosition:1
      >
1064
1065 Recovered from error: Unexpected token <Type:IDENT
      , Lexeme:a, BeginPosition:61, LinePosition:1> with
      lexeme 'a' at position 61 on line 1.
1066
1067
1068 DONE ALL EXPRESSION ATOMS
1069 OPERATOR STACK: deque([])
1070
1071 OUTPUT STACK: deque([AssignStatementNode (type::= 'INT',
      name::= 'a', left:: ='IdentifierNode (type
      ::= 'IDENT', name::= 'a', value:: =
      ExpressionStatementNode (type::= 'INT', name::= '10
      ', value:: ='IntegerLiteralNode (type::= 'INT',
      name::= '10', value:: ='10'))')operator:: ='=')
      right:: ='ExpressionStatementNode (type::= 'INT',
      name::= '10', value:: ='IntegerLiteralNode (type
      ::= 'INT', name::= '10', value:: ='10'))value
      :: ='ExpressionStatementNode (type::= 'INT', name
      ::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
      name::= '10', value:: ='10'))])
1072
1073
1074 CLEAR OPS STACK
1075 OPERATOR STACK deque([])
1076
1077
1078 OUTPUT STACK deque([AssignStatementNode (type::= 'INT',
      name::= 'a', left:: ='IdentifierNode (type
      ::= 'IDENT', name::= 'a', value:: =
      ExpressionStatementNode (type::= 'INT', name::= '10
      ', value:: ='IntegerLiteralNode (type::= 'INT',
      name::= '10', value:: ='10'))')operator:: ='=')
      right:: ='ExpressionStatementNode (type::= 'INT',
      name::= '10', value:: ='IntegerLiteralNode (type
```

```

1078 ::= 'INT', name::= '10', value:: ='10'))')value
      :: ='ExpressionStatementNode (type::= 'INT', name
      ::= '10', value:: ='IntegerLiteralNode (type::= '
      INT', name::= '10', value:: ='10'))'])
1079
1080
1081 This is the expr
1082
1083 AssignStatementNode (type::= 'INT', name::= 'a',
      left:: ='IdentifierNode (type::= 'IDENT', name::= '
      a', value:: ='ExpressionStatementNode (type::= 'INT
      ', name::= '10', value:: ='IntegerLiteralNode (type
      ::= 'INT', name::= '10', value:: ='10'))')
      operator:: ='=')right:: ='ExpressionStatementNode (
      type::= 'INT', name::= '10', value:: =
      IntegerLiteralNode (type::= 'INT', name::= '10',
      value:: ='10'))value:: ='ExpressionStatementNode
      (type::= 'INT', name::= '10', value:: =
      IntegerLiteralNode (type::= 'INT', name::= '10',
      value:: ='10'))')
1084
1085 <Type:;, Lexeme:;, BeginPosition:70, LinePosition:1
      >
1086
1087
1088 Main parser
1089 Working from => <Type:EOF, Lexeme:, BeginPosition:
      72, LinePosition:1> then False
1090
1091
1092 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value:: =deque([IdentifierNode (type::= 'IDENT', name::= 'a', value:: ='None'),
      IdentifierNode (type::= 'IDENT', name::= 'b', value
      :: ='None'), IdentifierNode (type::= 'IDENT', name
      ::= 'c', value:: ='None')])), AssignStatementNode
      (type::= 'INT', name::= 'a', left:: =
      IdentifierNode (type::= 'IDENT', name::= 'a', value
      :: ='ExpressionStatementNode (type::= 'INT', name
      ::= '10', value:: ='IntegerLiteralNode (type::= '
      INT', name::= '10', value:: ='10'))'))operator

```

```

1092 :: ='=')right:: ='ExpressionStatementNode (type
 ::= 'INT', name::= '10', value:: =
 IntegerLiteralNode (type::= 'INT', name::= '10',
 value:: ='10'))')value:: ='ExpressionStatementNode
 (type::= 'INT', name::= '10', value:: =
 IntegerLiteralNode (type::= 'INT', name::= '10',
 value:: ='10'))'), AssignStatementNode (type::= '
 INT', name::= 'b', left:: ='IdentifierNode (type
 ::= 'IDENT', name::= 'b', value:: =
 ExpressionStatementNode (type::= 'INT', name::= '5
 ', value:: ='IntegerLiteralNode (type::= 'INT',
 name::= '5', value:: ='5'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
 name::= '5', value:: ='IntegerLiteralNode (type
 ::= 'INT', name::= '5', value:: ='5'))')value:: =
 ExpressionStatementNode (type::= 'INT', name::= '5
 ', value:: ='IntegerLiteralNode (type::= 'INT',
 name::= '5', value:: ='5'))'), LetStatementNode (
 type::= 'IDENT', name::= '_d_e', value:: ='deque([
 IdentifierNode (type::= 'IDENT', name::= 'd', value
 :: ='None'), IdentifierNode (type::= 'IDENT', name
 ::= 'e', value:: ='None')]))'), AssignStatementNode
 (type::= 'INT', name::= 'c', left:: =
 IdentifierNode (type::= 'IDENT', name::= 'c', value
 :: ='ExpressionStatementNode (type::= 'INT', name
 ::= '10', value:: ='AssignStatementNode (type::= '
 INT', name::= 'a', left:: ='IdentifierNode (type
 ::= 'IDENT', name::= 'a', value:: =
 ExpressionStatementNode (type::= 'INT', name::= '10
 ', value:: ='IntegerLiteralNode (type::= 'INT',
 name::= '10', value:: ='10'))')')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
 name::= '10', value:: ='IntegerLiteralNode (type
 ::= 'INT', name::= '10', value:: ='10'))')value
 :: ='ExpressionStatementNode (type::= 'INT', name
 ::= '10', value:: ='IntegerLiteralNode (type::= '
 INT', name::= '10', value:: ='10'))')')')
operator:: ='=')right:: ='ExpressionStatementNode (
 type::= 'INT', name::= '10', value:: =
 AssignStatementNode (type::= 'INT', name::= 'a',
 left:: ='IdentifierNode (type::= 'IDENT', name::= '

```

```
1092 a', value:: ='ExpressionStatementNode (type::= 'INT
', name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: ='ExpressionStatementNode
(type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::=
'IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::=
'INT', name::= '10', value:: ='10'))'), 
AssignStatementNode (type::= 'INT', name::= 'd',
left:: ='IdentifierNode (type::= 'IDENT', name::=
'd', value:: =ExpressionStatementNode (type::=
'INT ', name::= '10', value:: =AssignStatementNode
(type::= 'INT', name::= 'c', left:: ='IdentifierNode
(type::= 'IDENT', name::= 'c', value:: =
ExpressionStatementNode (type::= 'INT', name::=
'10 ', value:: =AssignStatementNode (type::=
'INT', name::= 'a', left:: ='IdentifierNode (type::=
'IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::=
'10 ', value:: ='IntegerLiteralNode (type::=
'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::=
```

```

1092 INT', name::= '10', value:: ='10'))'))')))
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'a',
left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
value:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))value:: ='ExpressionStatementNode
(type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::= 'IDENT',
name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')')')')')')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT',
name::= 'c', left:: ='IdentifierNode (type::= 'IDENT',
name::= 'c', value:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='AssignStatementNode (
type::= 'INT', name::= 'a', left:: ='IdentifierNode
(type::= 'IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type

```

```
1092 ::= 'INT', name::= '10', value:: = '10'))')value
      :: ='ExpressionStatementNode (type::= 'INT', name
      ::= '10', value:: ='IntegerLiteralNode (type::= '
      INT', name::= '10', value:: ='10'))')')')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'a',
left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
value:: ='ExpressionStatementNode (type::= 'INT
', name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: ='ExpressionStatementNode
 (type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::= '
IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= '
INT', name::= '10', value:: ='10'))')')')value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='AssignStatementNode (type::= '
INT', name::= 'c', left:: ='IdentifierNode (type
::= 'IDENT', name::= 'c', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::= '
IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
```

```

1092 name::= '10', value::='10'))')operator::='=')
    right::='ExpressionStatementNode (type::= 'INT',
    name::= '10', value::='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value::='10'))value
    ::='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value::='IntegerLiteralNode (type::= '
    INT', name::= '10', value::='10'))'))')
operator::='=')right::='ExpressionStatementNode (
    type::= 'INT', name::= '10', value::=
    AssignStatementNode (type::= 'INT', name::= 'a',
    left::='IdentifierNode (type::= 'IDENT', name::= 'a',
    value::='ExpressionStatementNode (type::= 'INT',
    name::= '10', value::='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value::='10'))'))
operator::='=')right::='ExpressionStatementNode (
    type::= 'INT', name::= '10', value::=
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value::='10'))')value::='ExpressionStatementNode
    (type::= 'INT', name::= '10', value::=
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value::='10'))')value::=
    ExpressionStatementNode (type::= 'INT', name::= '10',
    value::='AssignStatementNode (type::= 'INT',
    name::= 'a', left::='IdentifierNode (type::= '
    IDENT', name::= 'a', value::=
    ExpressionStatementNode (type::= 'INT', name::= '10',
    value::='IntegerLiteralNode (type::= 'INT',
    name::= '10', value::='10'))')operator::='=')
right::='ExpressionStatementNode (type::= 'INT',
    name::= '10', value::='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value::='10'))')value
    ::='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value::='IntegerLiteralNode (type::= '
    INT', name::= '10', value::='10'))'))'),
    AssignStatementNode (type::= 'INT', name::= 'e',
    left::='IdentifierNode (type::= 'IDENT', name::= 'e',
    value::='ExpressionStatementNode (type::= 'INT',
    name::= '10', value::='AssignStatementNode (
    type::= 'INT', name::= 'a', left::='IdentifierNode
    (type::= 'IDENT', name::= 'a', value::=
    ExpressionStatementNode (type::= 'INT', name::= '10

```

```

1092 ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= '
    INT', name::= '10', value:: ='10'))')')')
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    AssignStatementNode (type::= 'INT', name::= 'a',
    left:: ='IdentifierNode (type::= 'IDENT', name::= '
        a', value:: ='ExpressionStatementNode (type::= 'INT
        ', name::= '10', value:: ='IntegerLiteralNode (type
        ::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))')value:: ='ExpressionStatementNode
    (type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))')')value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='AssignStatementNode (type::= 'INT',
    name::= 'a', left:: ='IdentifierNode (type::= '
        IDENT', name::= 'a', value:: =
        ExpressionStatementNode (type::= 'INT', name::= '10
        ', value:: ='IntegerLiteralNode (type::= 'INT',
        name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= '
    INT', name::= '10', value:: ='10'))')')'])
1093
1094
1095 HERE IS THE AST
1096 deque([LetStatementNode (type::= 'IDENT', name::= '_a_b_c', value:: ='deque([IdentifierNode (type::= 'IDENT', name::= 'a', value:: ='None'),
```

```

1096 IdentifierNode (type::= 'IDENT', name::= 'b', value
    :: = 'None'), IdentifierNode (type::= 'IDENT', name
    ::= 'c', value:: = 'None')]])), AssignStatementNode
    (type::= 'INT', name::= 'a', left:: =
IdentifierNode (type::= 'IDENT', name::= 'a', value
    :: = 'ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: = 'IntegerLiteralNode (type::= '
    INT', name::= '10', value:: = '10'))'))operator
    :: ='='right:: ='ExpressionStatementNode (type
    ::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: = '10'))')value:: ='ExpressionStatementNode
    (type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: = '10'))'), AssignStatementNode (type::= '
    INT', name::= 'b', left:: ='IdentifierNode (type
    ::= 'IDENT', name::= 'b', value:: =
ExpressionStatementNode (type::= 'INT', name::= '5
    ', value:: = 'IntegerLiteralNode (type::= 'INT',
    name::= '5', value:: = '5'))'))operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '5', value:: = 'IntegerLiteralNode (type
    ::= 'INT', name::= '5', value:: = '5'))')value:: =
ExpressionStatementNode (type::= 'INT', name::= '5
    ', value:: = 'IntegerLiteralNode (type::= 'INT',
    name::= '5', value:: = '5'))'), LetStatementNode (
    type::= 'IDENT', name::= '_d_e', value:: = 'deque([
IdentifierNode (type::= 'IDENT', name::= 'd', value
    :: = 'None'), IdentifierNode (type::= 'IDENT', name
    ::= 'e', value:: = 'None')]))'), AssignStatementNode
    (type::= 'INT', name::= 'c', left:: =
IdentifierNode (type::= 'IDENT', name::= 'c', value
    :: = 'ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: = 'AssignStatementNode (type::= '
    INT', name::= 'a', left:: ='IdentifierNode (type
    ::= 'IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: = 'IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: = '10'))'))operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: = 'IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: = '10'))')

```

```

1096 ::= 'INT', name::= '10', value:: = '10'))')value
      :: ='ExpressionStatementNode (type::= 'INT', name
      ::= '10', value:: ='IntegerLiteralNode (type::= '
      INT', name::= '10', value:: ='10'))')')')
operator:: ='=')right:: ='ExpressionStatementNode (
      type::= 'INT', name::= '10', value:: =
      AssignStatementNode (type::= 'INT', name::= 'a',
      left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
      value:: ='ExpressionStatementNode (type::= 'INT
      ', name::= '10', value:: ='IntegerLiteralNode (type
      ::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
      type::= 'INT', name::= '10', value:: =
      IntegerLiteralNode (type::= 'INT', name::= '10',
      value:: ='10'))')value:: ='ExpressionStatementNode
      (type::= 'INT', name::= '10', value:: =
      IntegerLiteralNode (type::= 'INT', name::= '10',
      value:: ='10'))')')value:: =
      ExpressionStatementNode (type::= 'INT', name::= '10
      ', value:: ='AssignStatementNode (type::= 'INT',
      name::= 'a', left:: ='IdentifierNode (type::= '
      IDENT', name::= 'a', value:: =
      ExpressionStatementNode (type::= 'INT', name::= '10
      ', value:: ='IntegerLiteralNode (type::= 'INT',
      name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
      name::= '10', value:: ='IntegerLiteralNode (type
      ::= 'INT', name::= '10', value:: ='10'))')value
      :: ='ExpressionStatementNode (type::= 'INT', name
      ::= '10', value:: ='IntegerLiteralNode (type::= '
      INT', name::= '10', value:: ='10'))')'),
      AssignStatementNode (type::= 'INT', name::= 'd',
      left:: ='IdentifierNode (type::= 'IDENT', name::= 'd',
      value:: ='ExpressionStatementNode (type::= 'INT
      ', name::= '10', value:: ='AssignStatementNode (
      type::= 'INT', name::= 'c', left:: ='IdentifierNode
      (type::= 'IDENT', name::= 'c', value:: =
      ExpressionStatementNode (type::= 'INT', name::= '10
      ', value:: ='AssignStatementNode (type::= 'INT',
      name::= 'a', left:: ='IdentifierNode (type::= '
      IDENT', name::= 'a', value:: =

```

```

1096 ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'a',
left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
value:: ='ExpressionStatementNode (type::= 'INT
', name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))value:: ='ExpressionStatementNode
(type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::= 'IDENT',
name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'c',
left:: ='IdentifierNode (type::= 'IDENT', name::= 'c',
value:: ='ExpressionStatementNode (type::= 'INT

```

```
1096 ', name::= '10', value:: = 'AssignStatementNode (
    type::= 'INT', name::= 'a', left:: = 'IdentifierNode
        (type::= 'IDENT', name::= 'a', value:: =
            ExpressionStatementNode (type::= 'INT', name::= '10
                ', value:: ='IntegerLiteralNode (type::= 'INT',
                    name::= '10', value:: ='10'))'))operator:: = '=')
right:: = 'ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
        ::= 'INT', name::= '10', value:: ='10'))')value
:: = 'ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::= '
        INT', name::= '10', value:: ='10'))'))')
operator:: = '=')right:: = 'ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
        AssignStatementNode (type::= 'INT', name::= 'a',
            left:: = 'IdentifierNode (type::= 'IDENT', name::= 'a',
                value:: = 'ExpressionStatementNode (type::= 'INT
                    ', name::= '10', value:: ='IntegerLiteralNode (type
                        ::= 'INT', name::= '10', value:: ='10'))'))
operator:: = '=')right:: = 'ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
        IntegerLiteralNode (type::= 'INT', name::= '10',
            value:: ='10'))')value:: = 'ExpressionStatementNode
        (type::= 'INT', name::= '10', value:: =
            IntegerLiteralNode (type::= 'INT', name::= '10',
                value:: ='10'))')')value:: =
        ExpressionStatementNode (type::= 'INT', name::= '10
            ', value:: = 'AssignStatementNode (type::= 'INT',
                name::= 'a', left:: = 'IdentifierNode (type::= '
                    IDENT', name::= 'a', value:: =
                        ExpressionStatementNode (type::= 'INT', name::= '10
                            ', value:: ='IntegerLiteralNode (type::= 'INT',
                                name::= '10', value:: ='10'))'))operator:: = '=')
right:: = 'ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
        ::= 'INT', name::= '10', value:: ='10'))')value
:: = 'ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::= '
        INT', name::= '10', value:: ='10'))'))')value
:: = 'ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: = 'AssignStatementNode (type::= '
```

```
1096 INT', name::= 'c', left::='IdentifierNode (type
    ::= 'IDENT', name::= 'c', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::=
    IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= '
INT', name::= '10', value:: ='10'))')')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
AssignStatementNode (type::= 'INT', name::= 'a',
left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
value:: ='ExpressionStatementNode (type::= 'INT
', name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')value:: ='ExpressionStatementNode
(type::= 'INT', name::= '10', value:: =
IntegerLiteralNode (type::= 'INT', name::= '10',
value:: ='10'))')')value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='AssignStatementNode (type::= 'INT',
name::= 'a', left:: ='IdentifierNode (type::=
    IDENT', name::= 'a', value:: =
ExpressionStatementNode (type::= 'INT', name::= '10
', value:: ='IntegerLiteralNode (type::= 'INT',
name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
name::= '10', value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10', value:: ='10'))value
:: ='ExpressionStatementNode (type::= 'INT', name
::= '10', value:: ='IntegerLiteralNode (type::= '
```

```

1096 INT', name::= '10', value:: ='10'))'))'))'))),
    AssignStatementNode (type::= 'INT', name::= 'e',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'e',
    value:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='AssignStatementNode (
    type::= 'INT', name::= 'a', left:: ='IdentifierNode
    (type::= 'IDENT', name::= 'a', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))value
    :: ='ExpressionStatementNode (type::= 'INT', name
    ::= '10', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')')')
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    AssignStatementNode (type::= 'INT', name::= 'a',
    left:: ='IdentifierNode (type::= 'IDENT', name::= 'a',
    value:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))')
operator:: ='=')right:: ='ExpressionStatementNode (
    type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))value:: ='ExpressionStatementNode
    (type::= 'INT', name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT', name::= '10',
    value:: ='10'))')')value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='AssignStatementNode (type::= 'INT',
    name::= 'a', left:: ='IdentifierNode (type::= 'IDENT',
    name::= 'a', value:: =
    ExpressionStatementNode (type::= 'INT', name::= '10
    ', value:: ='IntegerLiteralNode (type::= 'INT',
    name::= '10', value:: ='10'))')operator:: ='=')
right:: ='ExpressionStatementNode (type::= 'INT',
    name::= '10', value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10', value:: ='10'))')value
    :: ='ExpressionStatementNode (type::= 'INT', name
    
```

```

1096 ::= '10', value::='IntegerLiteralNode (type::= '
    INT', name::= '10', value::='10'))'))'])
1097
1098 (Parser)> sst
1099 +-----+-----+-----+
          +-----+
          +-----+
1100 | Name           | Type     | Line Declared   |
      Context Level | Value
          |
1101 +-----+-----+-----+-----+
          +-----+
          +-----+
1102 | Monke_beta_0.1 | None     | None:None
          |             0 | StatementListNode (type
          ::= 'STATEMENTS', name::=           |
1103 |             |         |           |
          |             | 'STATEMENTS', value::='deque([
          LetStatementNode (type::=           |
1104 |             |         |           |
          |             |             | 'IDENT', name::= '_a_b_c', value::='
          deque([IdentifierNode   |
1105 |             |         |           |
          |             |             | (type::= 'IDENT', name::= 'a', value
          ::= 'None'),           |
1106 |             |         |           |
          |             |             | IdentifierNode (type::= 'IDENT', name
          ::= 'b', value::=           |
1107 |             |         |           |
          |             |             | = 'None'), IdentifierNode (type::= '|
          IDENT', name::= 'c',           |
1108 |             |         |           |
          |             |             | value::='None'))']),
          AssignStatementNode (type::= 'INT',   |
1109 |             |         |           |
          |             |             | name::= 'a', left::= 'IdentifierNode (
          type::= 'IDENT',           |
1110 |             |         |           |
          |             |             | name::= 'a', value::='
          ExpressionStatementNode (type::=       |

```

```

1111 |           |           |           |
| 'INT', name::= '10', value:: =
IntegerLiteralNode (type::=   |
1112 |           |           |           |
| 'INT', name::= '10', value:: = '10
')'))'))operator:::
1113 |           |           |           |
| '=' )right:: ='ExpressionStatementNode
(type::= 'INT',   |
1114 |           |           |           |
| name::= '10', value:: =
IntegerLiteralNode (type::= 'INT',   |
1115 |           |           |           |
| name::= '10', value:: = '10'))')value
:::
1116 |           |           |           |
| = 'ExpressionStatementNode (type::= '
INT', name::= '10',   |
1117 |           |           |           |
| value:: = 'IntegerLiteralNode (type
::= 'INT', name::= '10',   |
1118 |           |           |           |
| value:: = '10'))',
AssignStatementNode (type::= 'INT',   |
1119 |           |           |           |
| name::= 'b', left:: ='IdentifierNode (
type::= 'IDENT',   |
1120 |           |           |           |
| name::= 'b', value:: =
ExpressionStatementNode (type::=   |
1121 |           |           |           |
| 'INT', name::= '5', value:: =
IntegerLiteralNode (type::=   |
1122 |           |           |           |
| 'INT', name::= '5', value:: = '5
'))'))operator:::
1123 |           |           |           |
| '=' )right:: ='ExpressionStatementNode
(type::= 'INT',   |
1124 |           |           |           |
| name::= '5', value:: =

```

```

1124 IntegerLiteralNode (type::= 'INT',      |
1125 |           |           |           |
1126 |           |           |           |
1127 |           |           |           |
1128 |           |           |           |
1129 |           |           |           |
1130 |           |           |           |
1131 |           |           |           |
1132 |           |           |           |
1133 |           |           |           |
1134 |           |           |           |
1135 |           |           |           |
1136 |           |           |           |
1137 |           |           |           |
1138 |           |           |           |

```

```
1138           | '10', value:: ='IntegerLiteralNode (|  
1139 |           type::= 'INT', name::= |  
1140 |           | '10', value:: ='10')'))'operator:: := '=')right:: |  
1141 |           | = 'ExpressionStatementNode (type::= '|  
1142 |           INT', name::= '10', |  
1143 |           | value:: ='IntegerLiteralNode (type::= '|  
1144 |           INT', name::= '10', value:: ='10'|  
1145 |           ')')')')')operator:: |  
1146 |           | = '=')right:: ='ExpressionStatementNode(|  
1147 |           type::= 'INT', |  
1148 |           | name::= '10', value:: ='|  
1149 |           AssignStatementNode (type::= 'INT', |  
1150 |           | name::= 'a', left:: ='IdentifierNode (|  
1151 |           type::= 'IDENT', |  
1152 |           | name::= 'a', value:: ='|  
1153 |           ExpressionStatementNode (type::= |  
1154 |           | 'INT', name::= '10', value:: ='10'|  
1155 |           ')')')')operator:: |  
1156 |           | = '=')right:: ='ExpressionStatementNode(|  
1157 |           type::= 'INT', |
```

```

1152 |           |           |
|       | name::= '10', value:: =
|       IntegerLiteralNode (type::= 'INT',   |
1153 |           |           |
|       | name::= '10', value:: ='10'))')value
|       ::           |
1154 |           |           |
|       | =ExpressionStatementNode (type::= '|
|       INT', name::= '10',      |
1155 |           |           |
|       | value:: ='IntegerLiteralNode (type
|       ::= 'INT', name::= '10',      |
1156 |           |           |
|       | value:: ='10'))')')')value:: =
|       ExpressionStatementNode   |
1157 |           |           |
|       | (type::= 'INT', name::= '10', value
|       :: ='AssignStatementNode   |
1158 |           |           |
|       | (type::= 'INT', name::= 'a', left:: =
|       IdentifierNode          |
1159 |           |           |
|       | (type::= 'IDENT', name::= 'a', value
|       ::           |
1160 |           |           |
|       | =ExpressionStatementNode (type::= '|
|       INT', name::= '10',      |
1161 |           |           |
|       | value:: ='IntegerLiteralNode (type
|       ::= 'INT', name::= '10',      |
1162 |           |           |
|       | value:: ='10'))')operator:: ='=')
|       right::           |
1163 |           |           |
|       | =ExpressionStatementNode (type::= '|
|       INT', name::= '10',      |
1164 |           |           |
|       | value:: ='IntegerLiteralNode (type
|       ::= 'INT', name::= '10',      |
1165 |           |           |
|       | value:: ='10'))')value:: =

```

```
1165 ExpressionStatementNode (type::= |
1166 |           |           |
| 'INT', name::= '10', value:: =' 
  IntegerLiteralNode (type::= |
1167 |           |           |
| 'INT', name::= '10', value:: ='10
 ')'))'))),
1168 |           |           |
| AssignStatementNode (type::= 'INT',
  name::= 'd', left:: |
1169 |           |           |
| ='IdentifierNode (type::= 'IDENT',
  name::= 'd', value:: |
1170 |           |           |
| ='ExpressionStatementNode (type::= ' 
  INT', name::= '10',      |
1171 |           |           |
| value:: ='AssignStatementNode (type
 ::= 'INT', name::= 'c',      |
1172 |           |           |
| left:: ='IdentifierNode (type::= ' 
  IDENT', name::= 'c',      |
1173 |           |           |
| value:: ='ExpressionStatementNode (
  type::= 'INT', name::= |
1174 |           |           |
| '10', value:: ='AssignStatementNode (
  type::= 'INT', name::= |
1175 |           |           |
| 'a', left:: ='IdentifierNode (type
 ::= 'IDENT', name::= 'a',      |
1176 |           |           |
| value:: ='ExpressionStatementNode (
  type::= 'INT', name::= |
1177 |           |           |
| '10', value:: ='IntegerLiteralNode (
  type::= 'INT', name::= |
1178 |           |           |
| '10', value:: ='10'))'))operator
 :: ='=')right:: |
1179 |           |           |
|
```

```
1179           | ='ExpressionStatementNode (type::= '
    INT', name::= '10',      |
1180   |           |           |
1181   |           | value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10',      |
1182   |           |           |
1183   |           |           | value:: ='10')')')value:: =
    ExpressionStatementNode (type::= |
1184   |           |           | 'INT', name::= '10', value:: ='|
    IntegerLiteralNode (type::=     |
1185   |           |           | 'INT', name::= '10', value:: ='10
    ')')')')operator::     |
1186   |           |           | '=')right:: ='ExpressionStatementNode
    (type::= 'INT',      |
1187   |           |           | name::= '10', value:: =
    AssignStatementNode (type::= 'INT',     |
1188   |           |           | name::= 'a', left:: ='IdentifierNode (
    type::= 'IDENT',      |
1189   |           |           | name::= 'a', value:: =
    ExpressionStatementNode (type::=     |
1190   |           |           | 'INT', name::= '10', value:: ='|
    IntegerLiteralNode (type::=     |
1191   |           |           | 'INT', name::= '10', value:: ='10
    ')')')operator::     |
1192   |           |           | '=')right:: ='ExpressionStatementNode
    (type::= 'INT',      |
1193   |           |           | name::= '10', value:: =
    IntegerLiteralNode (type::= 'INT',     |
1194   |           |           | name::= '10', value:: ='10')')')value
    ::
```

```
1193 |           |           |           |
      | = 'ExpressionStatementNode (type::= '
      |   INT', name::= '10',           |
1194 |           |           |           |
      |   value::= 'IntegerLiteralNode (type
      |   ::= 'INT', name::= '10',           |
1195 |           |           |           |
      |   value::= '10')'')')')')')value::=
      | ExpressionStatementNode           |
1196 |           |           |           |
      |   (type::= 'INT', name::= '10', value
      |   ::= 'AssignStatementNode           |
1197 |           |           |           |
      |   (type::= 'INT', name::= 'a', left::=
      | IdentifierNode           |
1198 |           |           |           |
      |   (type::= 'IDENT', name::= 'a', value
      |   ::=
1199 |           |           |           |
      |   value::= 'ExpressionStatementNode (type::= '
      |   INT', name::= '10',           |
1200 |           |           |           |
      |   value::= 'IntegerLiteralNode (type
      |   ::= 'INT', name::= '10',           |
1201 |           |           |           |
      |   value::= '10')'')')operator::= '=')
      | right:::
1202 |           |           |           |
      |   value::= 'ExpressionStatementNode (type::= '
      |   INT', name::= '10',           |
1203 |           |           |           |
      |   value::= 'IntegerLiteralNode (type
      |   ::= 'INT', name::= '10',           |
1204 |           |           |           |
      |   value::= '10')'')')value::=
      | ExpressionStatementNode (type::= |
1205 |           |           |           |
      |   'INT', name::= '10', value::=
      | IntegerLiteralNode (type::= |
1206 |           |           |           |
      |   'INT', name::= '10', value::= '10
```



```
1220           | value::='IntegerLiteralNode (type
  ::= 'INT', name::= '10',   |
1221 |           |           |
  | value::='10')'')')')')operator
  ::='=')right:::      |
1222 |           |           |
  | = 'ExpressionStatementNode (type::= '
    INT', name::= '10',   |
1223 |           |           |
  | value::='AssignStatementNode (type
  ::= 'INT', name::= 'a',   |
1224 |           |           |
  | left::='IdentifierNode (type::= '
    IDENT', name::= 'a',   |
1225 |           |           |
  | value::='ExpressionStatementNode (
    type::= 'INT', name::=   |
1226 |           |           |
  | '10', value::='IntegerLiteralNode (
    type::= 'INT', name::=   |
1227 |           |           |
  | '10', value::='10')'')')')operator
  ::='=')right:::      |
1228 |           |           |
  | = 'ExpressionStatementNode (type::= '
    INT', name::= '10',   |
1229 |           |           |
  | value::='IntegerLiteralNode (type
  ::= 'INT', name::= '10',   |
1230 |           |           |
  | value::='10')')')value::='
  ExpressionStatementNode (type::= |
1231 |           |           |
  | 'INT', name::= '10', value::='
  IntegerLiteralNode (type::=   |
1232 |           |           |
  | 'INT', name::= '10', value::='10
  ')')')')')value::      |
1233 |           |           |
  | = 'ExpressionStatementNode (type::= '
    INT', name::= '10',   |
```

```

1234 |           |           |           |
      | value::= 'AssignStatementNode (type
      ::= 'INT', name::= 'a',   |
1235 |           |           |           |
      | left::= 'IdentifierNode (type::= '
      IDENT', name::= 'a',     |
1236 |           |           |           |
      | value::= 'ExpressionStatementNode (
      type::= 'INT', name::=   |
1237 |           |           |           |
      | '10', value::= 'IntegerLiteralNode (
      type::= 'INT', name::=   |
1238 |           |           |           |
      | '10', value::= '10')'))')operator
      ::= '=')right:::       |
1239 |           |           |           |
      | = 'ExpressionStatementNode (type::= '
      INT', name::= '10',     |
1240 |           |           |           |
      | value::= 'IntegerLiteralNode (type
      ::= 'INT', name::= '10',   |
1241 |           |           |           |
      | value::= '10')')')value::= '
      ExpressionStatementNode (type::= |
1242 |           |           |           |
      | 'INT', name::= '10', value::= '
      IntegerLiteralNode (type::=   |
1243 |           |           |           |
      | 'INT', name::= '10', value::= '10
      ')')')')')')')value:::    |
1244 |           |           |           |
      | = 'ExpressionStatementNode (type::= '
      INT', name::= '10',     |
1245 |           |           |           |
      | value::= 'AssignStatementNode (type
      ::= 'INT', name::= 'c',   |
1246 |           |           |           |
      | left::= 'IdentifierNode (type::= '
      IDENT', name::= 'c',     |
1247 |           |           |           |
      | value::= 'ExpressionStatementNode (

```

```
1247 type ::= 'INT', name ::= |  
1248 | | |  
1249 | | | '10', value ::= 'AssignStatementNode ('  
| | | type ::= 'INT', name ::= |  
1249 | | | |  
1250 | | | | 'a', left ::= 'IdentifierNode (type  
| | | | ::= 'IDENT', name ::= 'a', |  
1250 | | | | |  
1251 | | | | | value ::= 'ExpressionStatementNode ('  
| | | | | type ::= 'INT', name ::= |  
1251 | | | | | |  
1252 | | | | | | | '10', value ::= 'IntegerLiteralNode ('  
| | | | | | | type ::= 'INT', name ::= |  
1252 | | | | | | | |  
1253 | | | | | | | | '| '10', value ::= ='10')'))'))operator  
| | | | | | | | :: ='=')right ::= |  
1253 | | | | | | | | |  
1254 | | | | | | | | | value ::= 'ExpressionStatementNode (type ::= '  
| | | | | | | | | INT', name ::= '10', |  
1254 | | | | | | | | | |  
1255 | | | | | | | | | | value ::= 'IntegerLiteralNode (type  
| | | | | | | | | | ::= 'INT', name ::= '10', |  
1255 | | | | | | | | | | |  
1256 | | | | | | | | | | | value ::= ='10')'))value ::= '  
| | | | | | | | | | | ExpressionStatementNode (type ::= |  
1256 | | | | | | | | | | | |  
1257 | | | | | | | | | | | | | 'INT', name ::= '10', value ::= ='10  
| | | | | | | | | | | | | | '| ')'))'))'))operator ::= |  
1257 | | | | | | | | | | | | | | |  
1258 | | | | | | | | | | | | | |= ='=')right ::= 'ExpressionStatementNode  
| | | | | | | | | | | | | | (type ::= 'INT', |  
1258 | | | | | | | | | | | | | | | |  
1259 | | | | | | | | | | | | | | | | | name ::= '10', value ::= '  
| | | | | | | | | | | | | | | | | AssignStatementNode (type ::= 'INT', |  
1259 | | | | | | | | | | | | | | | | | |  
1260 | | | | | | | | | | | | | | | | | | | name ::= 'a', left ::= 'IdentifierNode ('  
| | | | | | | | | | | | | | | | | | | type ::= 'IDENT', |  
1260 | | | | | | | | | | | | | | | | | | | |  
1261 | | | | | | | | | | | | | | | | | | | | |
```

```
1261           | name::= 'a', value::='
    ExpressionStatementNode (type::=
1262   |           |           |
           | 'INT', name::= '10', value::='
    IntegerLiteralNode (type::=
1263   |           |           |
           | 'INT', name::= '10', value::='10
     ')')')operator::=
1264   |           |           |
           | '='right::='ExpressionStatementNode
     (type::= 'INT',
1265   |           |           |
           | name::= '10', value::='
    IntegerLiteralNode (type::= 'INT',
1266   |           |           |
           | name::= '10', value::='10')')value
     ::=
1267   |           |           |
           | ='ExpressionStatementNode (type::=
     'INT', name::= '10',
1268   |           |           |
           | value::='IntegerLiteralNode (type
     ::= 'INT', name::= '10',
1269   |           |           |
           | value::='10')')')')value::='
    ExpressionStatementNode
1270   |           |           |
           | (type::= 'INT', name::= '10', value
     ::='AssignStatementNode
1271   |           |           |
           | (type::= 'INT', name::= 'a', left::=
     IdentifierNode
1272   |           |           |
           | (type::= 'IDENT', name::= 'a', value
     ::=
1273   |           |           |
           | ='ExpressionStatementNode (type::=
     'INT', name::= '10',
1274   |           |           |
           | value::='IntegerLiteralNode (type
     ::= 'INT', name::= '10',
```

```
1275 |           |           |           |
|       | value::= '10')'))')operator::= '=')
|       right:::
1276 |           |           |           |
|       | = 'ExpressionStatementNode (type::= '
|       INT', name::= '10',           |
1277 |           |           |           |
|       | value::= 'IntegerLiteralNode (type
|       ::= 'INT', name::= '10',           |
1278 |           |           |           |
|       | value::= '10')'))value::=
|       ExpressionStatementNode (type::= |
1279 |           |           |           |
|       | 'INT', name::= '10', value::=
|       IntegerLiteralNode (type::=   |
1280 |           |           |           |
|       | 'INT', name::= '10', value::= '10
|       ')'))'))'))),
1281 |           |           |           |
|       | AssignStatementNode (type::= 'INT',
|       name::= 'e', left:::
1282 |           |           |           |
|       | = 'IdentifierNode (type::= 'IDENT',
|       name::= 'e', value:::
1283 |           |           |           |
|       | = 'ExpressionStatementNode (type::= '
|       INT', name::= '10',           |
1284 |           |           |           |
|       | value::= 'AssignStatementNode (type
|       ::= 'INT', name::= 'a',           |
1285 |           |           |           |
|       | left::= 'IdentifierNode (type::= '
|       IDENT', name::= 'a',           |
1286 |           |           |           |
|       | value::= 'ExpressionStatementNode (
|       type::= 'INT', name::=           |
1287 |           |           |           |
|       | '10', value::= 'IntegerLiteralNode (
|       type::= 'INT', name::=           |
1288 |           |           |           |
|       | '10', value::= '10')'))')operator
```

```
1288 ::='=')right::          |
1289 |           |           |
|           | = 'ExpressionStatementNode (type::= ' |
|           |           |   INT', name::= '10',       |
1290 |           |           |           |
|           |           | value::= 'IntegerLiteralNode (type |
|           |           |   ::= 'INT', name::= '10',       |
1291 |           |           |           |
|           |           | value::= '10'))')value::= ' |
|           |           | ExpressionStatementNode (type::= | |
1292 |           |           |           |   |
|           |           |           |   | 'INT', name::= '10', value::= ' |
|           |           |           | IntegerLiteralNode (type::= | |
1293 |           |           |           |           |   |
|           |           |           |           |   | 'INT', name::= '10', value::= '10 |
|           |           |           |           |   ')')')')operator::= |
1294 |           |           |           |           |   |
|           |           |           |           |   | = '=')right::= 'ExpressionStatementNode |
|           |           |           |           |   (type::= 'INT',       |
1295 |           |           |           |           |   |
|           |           |           |           |   | name::= '10', value::= ' |
|           |           |           |           | AssignStatementNode (type::= 'INT',   |
1296 |           |           |           |           |   |
|           |           |           |           |   | name::= 'a', left::= 'IdentifierNode ( |
|           |           |           |           |   type::= 'IDENT',       |
1297 |           |           |           |           |   |
|           |           |           |           |   | name::= 'a', value::= ' |
|           |           |           |           | ExpressionStatementNode (type::=       |
1298 |           |           |           |           |   |
|           |           |           |           |   |   'INT', name::= '10', value::= ' |
|           |           |           |           | IntegerLiteralNode (type::= | |
1299 |           |           |           |           |   |
|           |           |           |           |   |   'INT', name::= '10', value::= '10 |
|           |           |           |           |   ')')')operator::= |
1300 |           |           |           |           |   |
|           |           |           |           |   | = '=')right::= 'ExpressionStatementNode |
|           |           |           |           |   (type::= 'INT',       |
1301 |           |           |           |           |   |
|           |           |           |           |   | name::= '10', value::= ' |
|           |           |           |           | IntegerLiteralNode (type::= 'INT',   |
1302 |           |           |           |           |   |
```

```

1302           | name::= '10', value:: ='10')')')value
1303           ::           |
1303   |           |       |
1303   |           | = 'ExpressionStatementNode (type::= '
1304     INT', name::= '10',      |
1304   |           |       |
1304   |           | value:: ='IntegerLiteralNode (type
1304   ::= 'INT', name::= '10',      |
1305   |           |       |
1305   |           | value:: ='10')')')')')value:: =
1305     ExpressionStatementNode      |
1306   |           |       |
1306   |           | (type::= 'INT', name::= '10', value
1306   ::= 'AssignStatementNode      |
1307   |           |       |
1307   |           | (type::= 'INT', name::= 'a', left:: =
1307     IdentifierNode            |
1308   |           |       |
1308   |           | (type::= 'IDENT', name::= 'a', value
1308   ::           |
1309   |           |       |
1309   |           | = 'ExpressionStatementNode (type::= '
1309     INT', name::= '10',      |
1310   |           |       |
1310   |           | value:: ='IntegerLiteralNode (type
1310   ::= 'INT', name::= '10',      |
1311   |           |       |
1311   |           | value:: ='10')')')operator:: ='=')
1311     right::           |
1312   |           |       |
1312   |           | = 'ExpressionStatementNode (type::= '
1312     INT', name::= '10',      |
1313   |           |       |
1313   |           | value:: ='IntegerLiteralNode (type
1313   ::= 'INT', name::= '10',      |
1314   |           |       |
1314   |           | value:: ='10')')')value:: =
1314     ExpressionStatementNode (type::= |
1315   |           |       |
1315   |           | 'INT', name::= '10', value:: =
1315     IntegerLiteralNode (type::=      |

```

```

1316 |           |           |           |
|   'INT', name::= '10', value:: = '10
|   ')')')')])')
1317 +-----+-----+-----+
-----+
-----+
1318 | a           | None     | 15:1
|           |          0 |
ExpressionStatementNode (type::= 'INT', name::= '10
',
1319 |           |           |           |
|           | value:: ='IntegerLiteralNode (type
::= 'INT', name::= '10',      |
1320 |           |           |           |
|           | value:: ='10
|   ')')
1321 +-----+-----+-----+
-----+
-----+
1322 | b           | None     | 23:1
|           |          0 |
ExpressionStatementNode (type::= 'INT', name::= '5
', value:: |
1323 |           |           |           |
|           | ='IntegerLiteralNode (type::= 'INT',
name::= '5', value::    |
1324 |           |           |           |
|           | = '5
|   ')')
|
1325 +-----+-----+-----+
-----+
-----+
1326 | c           | None     | 15:1
|           |          0 |
ExpressionStatementNode (type::= 'INT', name::= '10
',
1327 |           |           |           |
|           | value:: ='AssignStatementNode (type
::= 'INT', name::= 'a',      |
1328 |           |           |           |
|           |          0 |

```

```

1328           | left:: ='IdentifierNode (type::= '
  IDENT', name::= 'a',          |
1329 |           |           |
  | value:: ='ExpressionStatementNode ('
    type::= 'INT', name::=   |
1330 |           |           |
  | '10', value:: ='IntegerLiteralNode ('
    type::= 'INT', name::=   |
1331 |           |           |
  | '10', value:: ='10')'))')operator
    :: ='=')right::           |
1332 |           |           |
  | = 'ExpressionStatementNode (type::= '
    INT', name::= '10',        |
1333 |           |           |
  | value:: ='IntegerLiteralNode (type
    ::= 'INT', name::= '10',   |
1334 |           |           |
  | value:: ='10')'))value:: =
  ExpressionStatementNode (type::= |
1335 |           |           |
  | 'INT', name::= '10', value:: =
  IntegerLiteralNode (type::=   |
1336 |           |           |
  | 'INT', name::= '10', value:: ='10
  ')'))')
1337 +-----+-----+-----+
-----+
-----+
1338 | d           | None    | 15:1
  |           | 0 |
  ExpressionStatementNode (type::= 'INT', name::= '10
  ',          |
1339 |           |           |
  | value:: ='AssignStatementNode (type
    ::= 'INT', name::= 'c',      |
1340 |           |           |
  | left:: ='IdentifierNode (type::= '
    IDENT', name::= 'c',       |
1341 |           |           |
  | value:: ='ExpressionStatementNode ('

```

```

1341 type ::= 'INT', name ::= | |
1342 | | |
| | '10', value ::= 'AssignStatementNode ( |
| type ::= 'INT', name ::= | |
1343 | | |
| | 'a', left ::= 'IdentifierNode (type |
| ::= 'IDENT', name ::= 'a', | |
1344 | | |
| | value ::= 'ExpressionStatementNode ( |
| type ::= 'INT', name ::= | |
1345 | | |
| | '10', value ::= 'IntegerLiteralNode ( |
| type ::= 'INT', name ::= | |
1346 | | |
| | '10', value ::= ='10')'))'))operator |
| :: ='=')right ::= | |
1347 | | |
| | | = 'ExpressionStatementNode (type ::= ' |
| INT', name ::= '10', | |
1348 | | |
| | | value ::= 'IntegerLiteralNode (type |
| ::= 'INT', name ::= '10', | |
1349 | | |
| | | value ::= ='10')'))value ::= ' |
| ExpressionStatementNode (type ::= | |
1350 | | |
| | | 'INT', name ::= '10', value ::= ' |
| IntegerLiteralNode (type ::= | |
1351 | | |
| | | 'INT', name ::= '10', value ::= ='10 |
| ')'))'))'))operator ::= | |
1352 | | |
| | | = ='=')right ::= 'ExpressionStatementNode |
| (type ::= 'INT', | |
1353 | | |
| | | name ::= '10', value ::= ' |
| AssignStatementNode (type ::= 'INT', | |
1354 | | |
| | | name ::= 'a', left ::= 'IdentifierNode ( |
| type ::= 'IDENT', | |
1355 | | |
| | | | |

```

```

1355           | name::= 'a', value::=
    ExpressionStatementNode (type::=      |
1356   |           |           |
1356   |           | 'INT', name::= '10', value::=
    IntegerLiteralNode (type::=      |
1357   |           |           |
1357   |           | 'INT', name::= '10', value::= '10
     ')')')operator::      |
1358   |           |           |
1358   |           | '=' )right:: ='ExpressionStatementNode
     (type::= 'INT',      |
1359   |           |           |
1359   |           | name::= '10', value::=
     IntegerLiteralNode (type::= 'INT',      |
1360   |           |           |
1360   |           | name::= '10', value::= '10')')')value
     ::      |
1361   |           |           |
1361   |           | = 'ExpressionStatementNode (type::= '
     INT', name::= '10',      |
1362   |           |           |
1362   |           | value::= 'IntegerLiteralNode (type
     ::= 'INT', name::= '10',      |
1363   |           |           |
1363   |           | value::= '10')')')')')value:: =
     ExpressionStatementNode      |
1364   |           |           |
1364   |           | (type::= 'INT', name::= '10', value
     ::= 'AssignStatementNode      |
1365   |           |           |
1365   |           | (type::= 'INT', name::= 'a', left::=
     IdentifierNode      |
1366   |           |           |
1366   |           | (type::= 'IDENT', name::= 'a', value
     ::      |
1367   |           |           |
1367   |           | = 'ExpressionStatementNode (type::= '
     INT', name::= '10',      |
1368   |           |           |
1368   |           | value::= 'IntegerLiteralNode (type
     ::= 'INT', name::= '10',      |

```

```

1369 |           |           |           |
|       value::= '10')')')operator::= '=')
|       right:::
1370 |           |           |           |
|       | = 'ExpressionStatementNode (type::= '
|       INT', name::= '10',           |
1371 |           |           |           |
|       |       value::= 'IntegerLiteralNode (type
|       ::= 'INT', name::= '10',           |
1372 |           |           |           |
|       |       value::= '10')')')value::=
|       ExpressionStatementNode (type::= |
1373 |           |           |           |
|       |       |       'INT', name::= '10', value::=
|       IntegerLiteralNode (type::=   |
1374 |           |           |           |
|       |       |       |       'INT', name::= '10', value::= '10
|       ')')')')')')
1375 +-----+-----+-----+
-----+
-----+
-----+
1376 | e           | None     | 15:1
|           |          0 |
|       ExpressionStatementNode (type::= 'INT', name::= '10
|       ',           |
1377 |           |           |           |
|       |       value::= 'AssignStatementNode (type
|       ::= 'INT', name::= 'a',           |
1378 |           |           |           |
|       |       |       left::= 'IdentifierNode (type::= '
|       IDENT', name::= 'a',           |
1379 |           |           |           |
|       |       |       value::= 'ExpressionStatementNode (
|       type::= 'INT', name::=   |
1380 |           |           |           |
|       |       |       |       '10', value::= 'IntegerLiteralNode (
|       type::= 'INT', name::=   |
1381 |           |           |           |
|       |       |       |       |       '10', value::= '10')')')operator
|       ::= '=')right:::
1382 |           |           |           |

```

```
1382          | ='ExpressionStatementNode (type::= '
  INT', name::= '10',      |
1383 |           |       |
  | value::='IntegerLiteralNode (type
  ::= 'INT', name::= '10',   |
1384 |           |       |
  | value::='10'))')value::='
  ExpressionStatementNode (type::= |
1385 |           |       |
  | 'INT', name::= '10', value::='
  IntegerLiteralNode (type::=   |
1386 |           |       |
  | 'INT', name::= '10', value::='10
  ')'))'))'
1387 +-----+-----+-----+
  -----+
  -----+
1388 (Parser)>
```