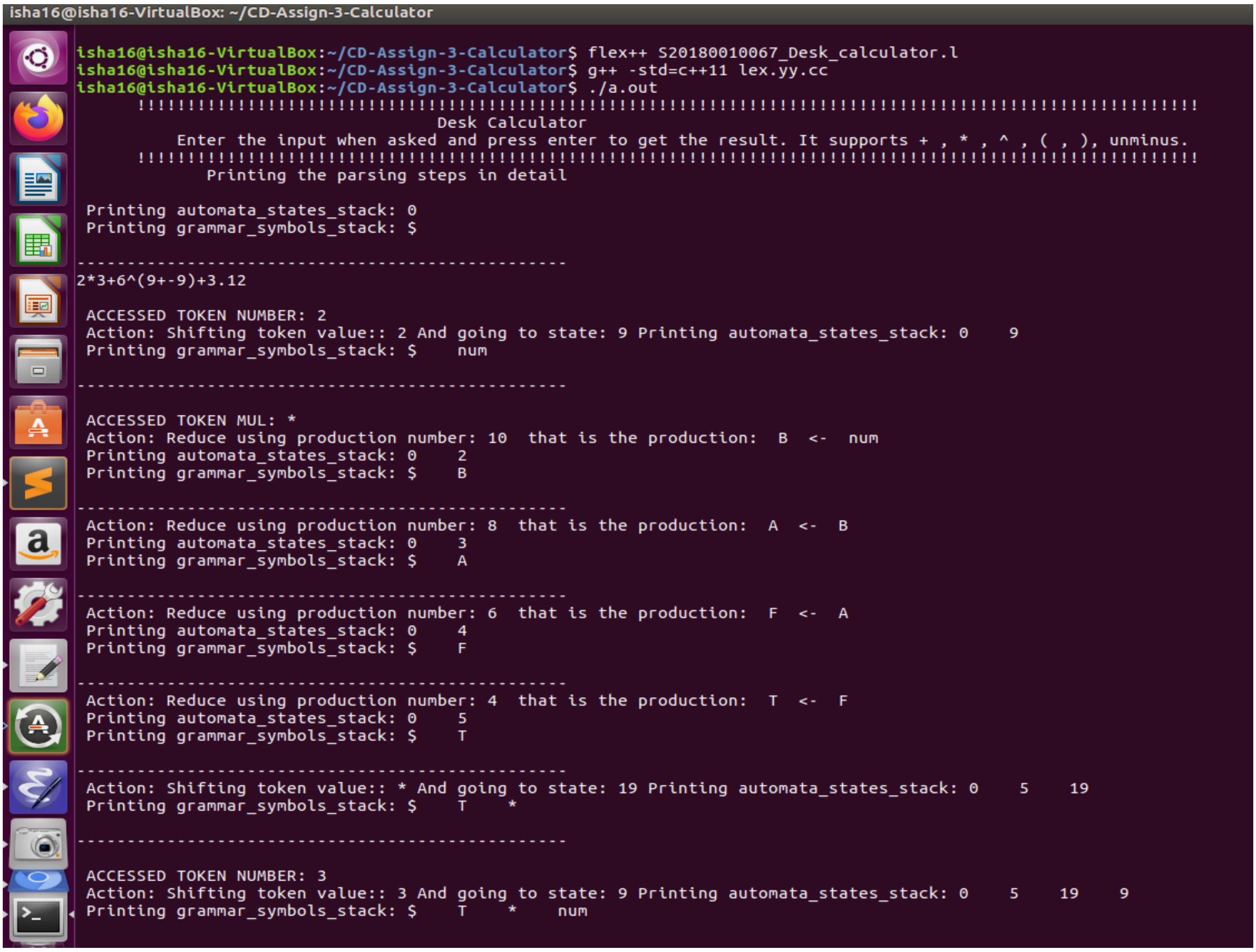
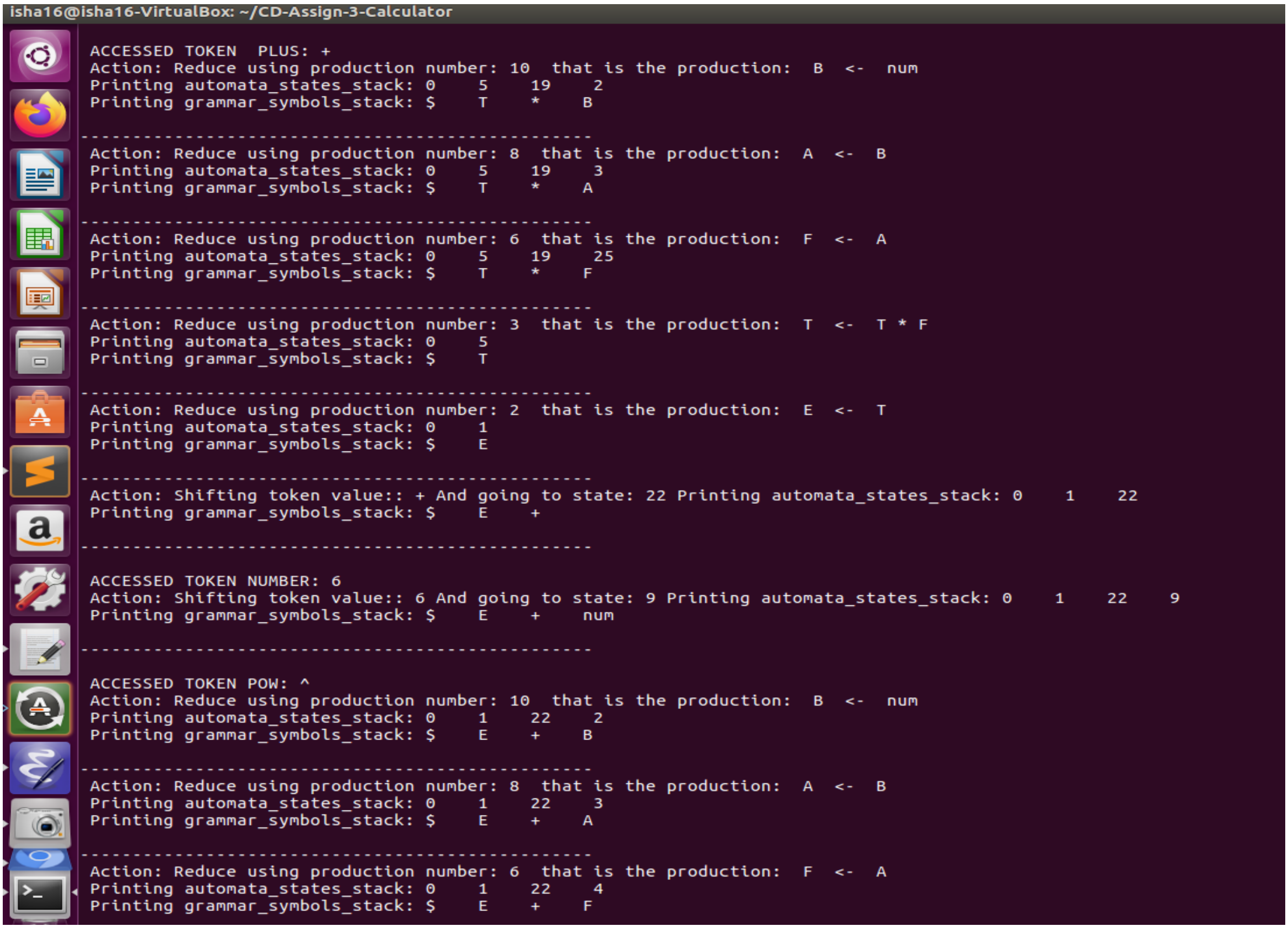
Desk Calculator Test Case screenshot



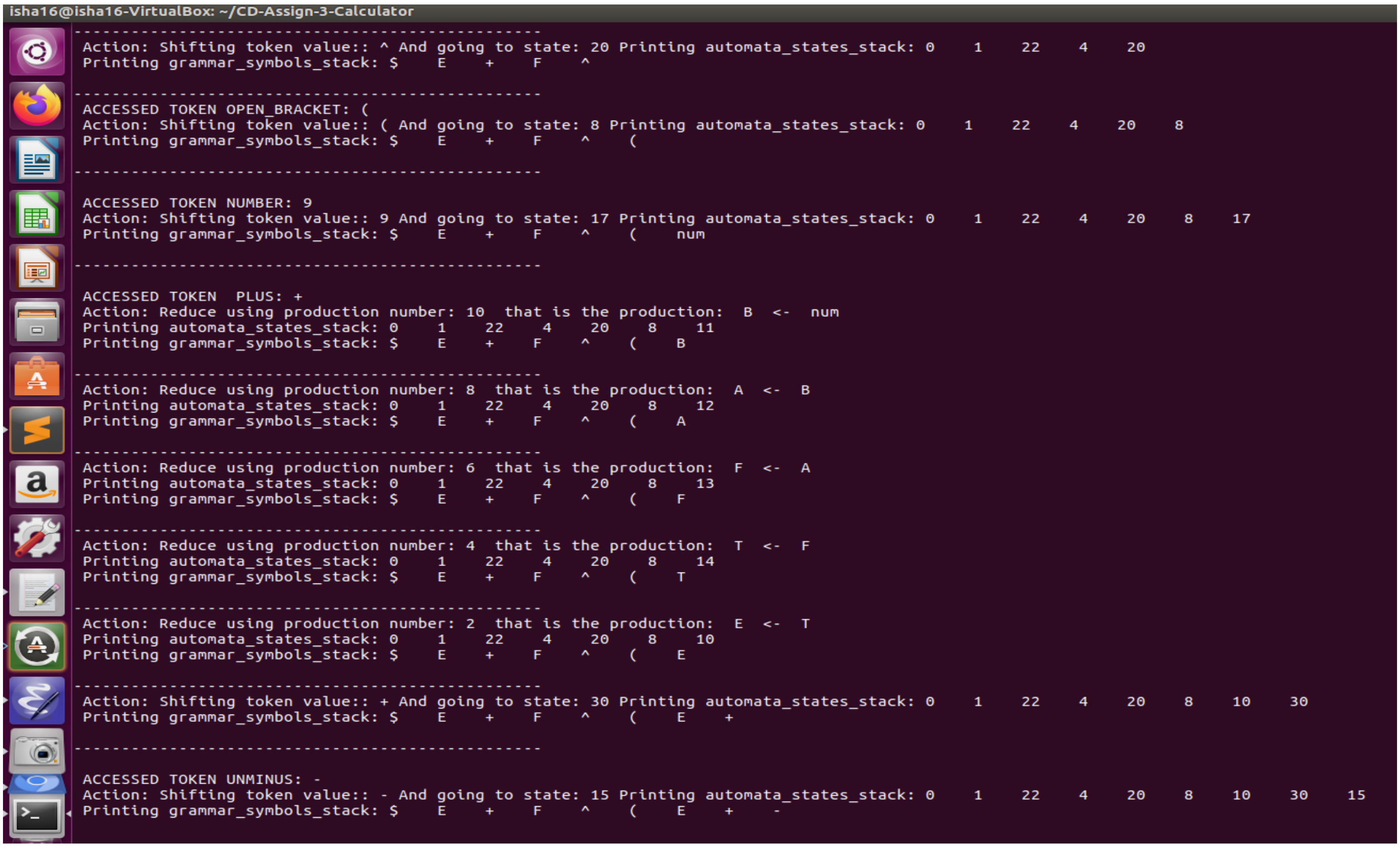
isha16@isha16-VirtualBox: ~/CD-Assign-3-Calculator

```
isha16@isha16-VirtualBox:~/CD-Assign-3-Calculator$ flex++ S20180010067_Desk_calculator.l
isha16@isha16-VirtualBox:~/CD-Assign-3-Calculator$ g++ -std=c++11 lex.yy.cc
isha16@isha16-VirtualBox:~/CD-Assign-3-Calculator$ ./a.out
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                                 Desk Calculator
          Enter the input when asked and press enter to get the result. It supports + , * , ^ , ( , ), unminus.
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
             Printing the parsing steps in detail

 Printing automata_states_stack: 0
 Printing grammar_symbols_stack: $

-------------------------------------------------
2*3+6^(9+-9)+3.12

 ACCESSED TOKEN NUMBER: 2
 Action: Shifting token value:: 2 And going to state: 9 Printing automata_states_stack: 0    9
 Printing grammar_symbols_stack: $    num

-------------------------------------------------

 ACCESSED TOKEN MUL: *
 Action: Reduce using production number: 10  that is the production:  B  <-  num
 Printing automata_states_stack: 0    2
 Printing grammar_symbols_stack: $    B

-------------------------------------------------
 Action: Reduce using production number: 8  that is the production:  A  <-  B
 Printing automata_states_stack: 0    3
 Printing grammar_symbols_stack: $    A

-------------------------------------------------
 Action: Reduce using production number: 6  that is the production:  F  <-  A
 Printing automata_states_stack: 0    4
 Printing grammar_symbols_stack: $    F

-------------------------------------------------
 Action: Reduce using production number: 4  that is the production:  T  <-  F
 Printing automata_states_stack: 0    5
 Printing grammar_symbols_stack: $    T

-------------------------------------------------
 Action: Shifting token value:: * And going to state: 19 Printing automata_states_stack: 0    5    19
 Printing grammar_symbols_stack: $    T    *

-------------------------------------------------

 ACCESSED TOKEN NUMBER: 3
 Action: Shifting token value:: 3 And going to state: 9 Printing automata_states_stack: 0    5    19    9
 Printing grammar_symbols_stack: $    T    *    num
```



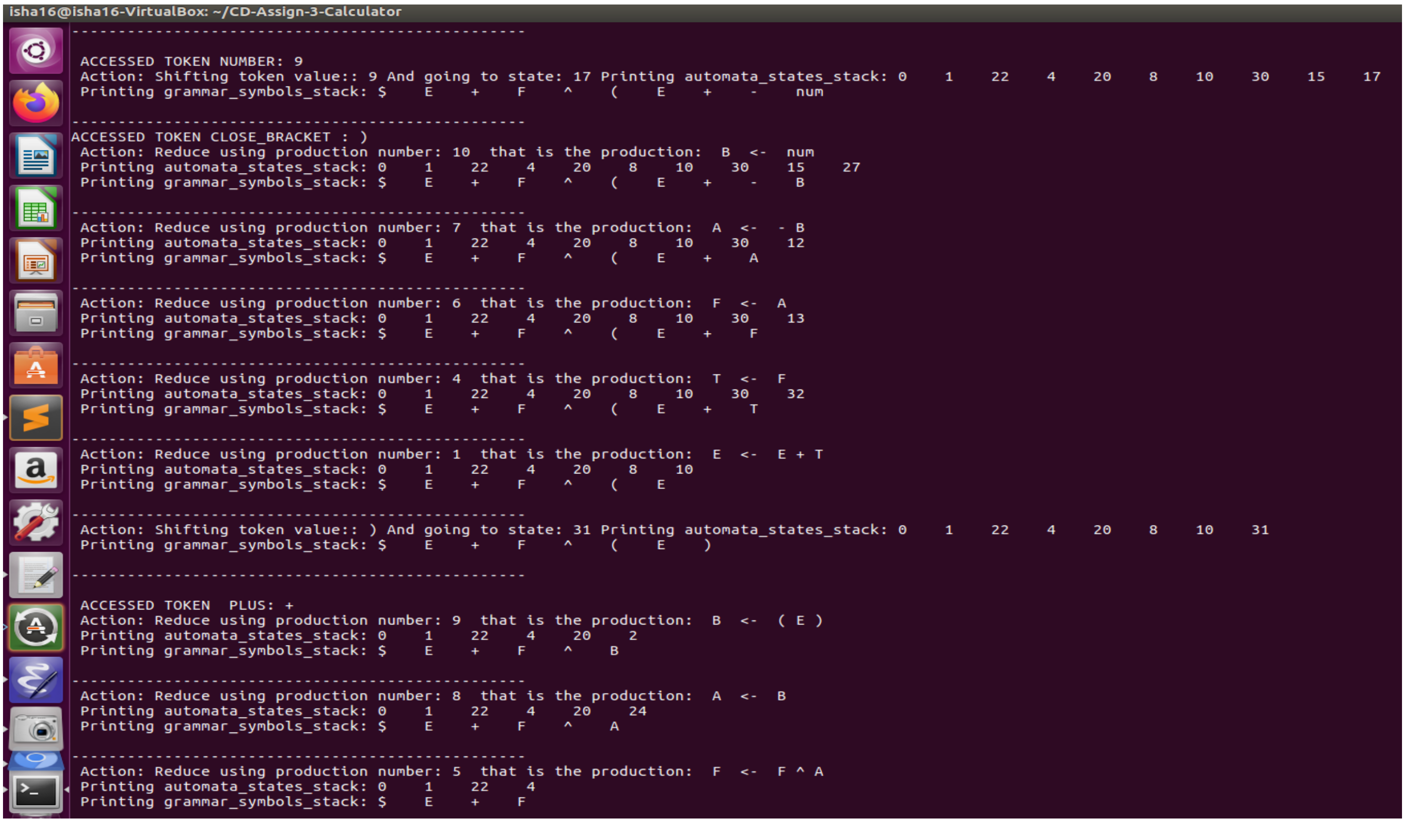
isha16@isha16-VirtualBox: ~/CD-Assign-3-Calculator

```
ACCESSED TOKEN  PLUS: +
Action: Reduce using production number: 10  that is the production:  B  <-  num
Printing automata_states_stack: 0    5    19    2
Printing grammar_symbols_stack: $    T    *    B

-----------------------------------------------
Action: Reduce using production number: 8  that is the production:  A  <-  B
Printing automata_states_stack: 0    5    19    3
Printing grammar_symbols_stack: $    T    *    A

-----------------------------------------------
Action: Reduce using production number: 6  that is the production:  F  <-  A
Printing automata_states_stack: 0    5    19    25
Printing grammar_symbols_stack: $    T    *    F

-----------------------------------------------
Action: Reduce using production number: 3  that is the production:  T  <-  T * F
Printing automata_states_stack: 0    5
Printing grammar_symbols_stack: $    T

-----------------------------------------------
Action: Reduce using production number: 2  that is the production:  E  <-  T
Printing automata_states_stack: 0    1
Printing grammar_symbols_stack: $    E

-----------------------------------------------
Action: Shifting token value:: + And going to state: 22 Printing automata_states_stack: 0    1    22
Printing grammar_symbols_stack: $    E    +

-----------------------------------------------

ACCESSED TOKEN NUMBER: 6
Action: Shifting token value:: 6 And going to state: 9 Printing automata_states_stack: 0    1    22    9
Printing grammar_symbols_stack: $    E    +    num

-----------------------------------------------

ACCESSED TOKEN POW: ^
Action: Reduce using production number: 10  that is the production:  B  <-  num
Printing automata_states_stack: 0    1    22    2
Printing grammar_symbols_stack: $    E    +    B

-----------------------------------------------
Action: Reduce using production number: 8  that is the production:  A  <-  B
Printing automata_states_stack: 0    1    22    3
Printing grammar_symbols_stack: $    E    +    A

-----------------------------------------------
Action: Reduce using production number: 6  that is the production:  F  <-  A
Printing automata_states_stack: 0    1    22    4
Printing grammar_symbols_stack: $    E    +    F
```



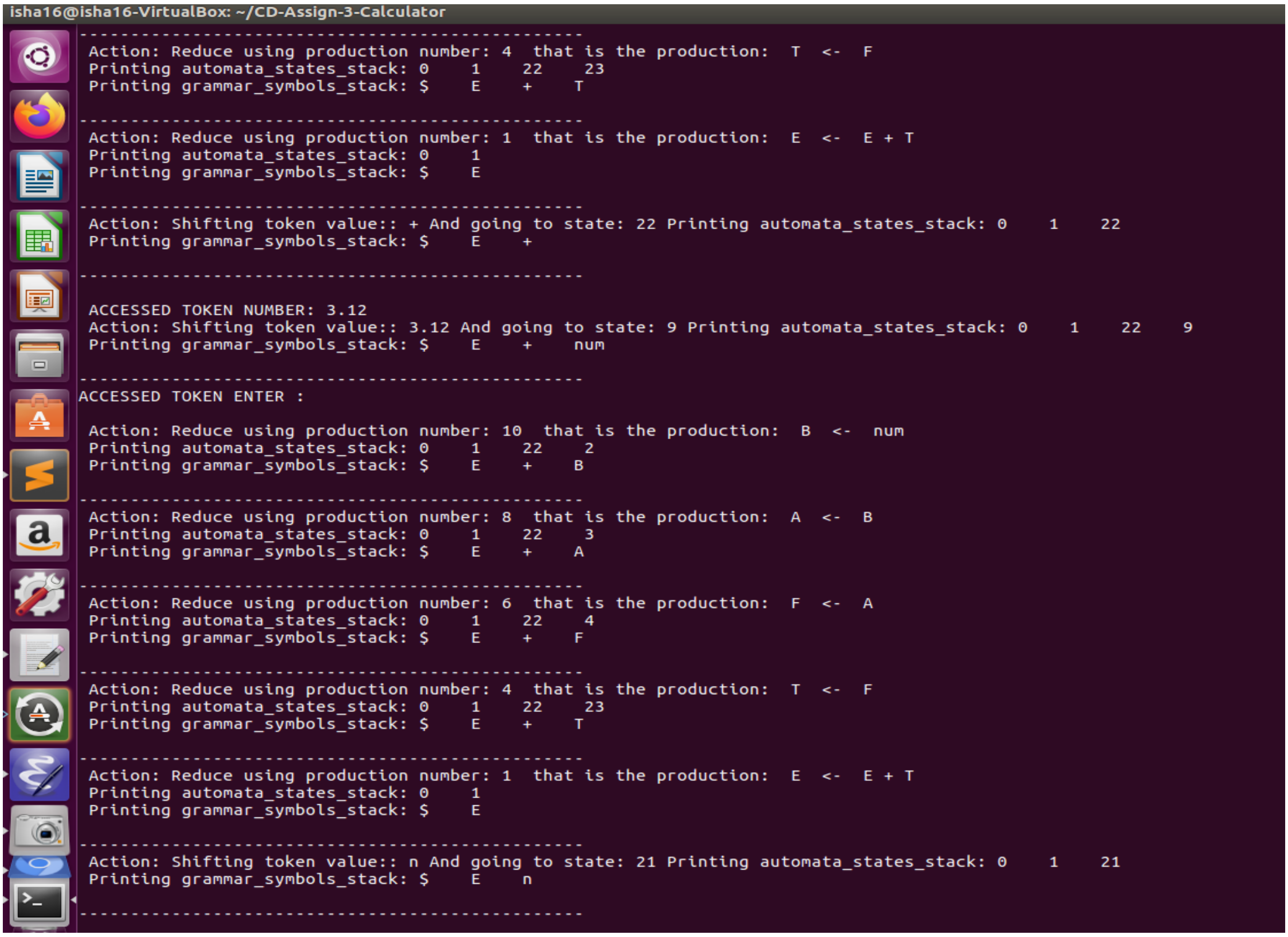
isha16@isha16-VirtualBox: ~/CD-Assign-3-Calculator

```
-------------------------------------------------
 Action: Shifting token value:: ^ And going to state: 20 Printing automata_states_stack: 0    1    22    4    20
 Printing grammar_symbols_stack: $    E    +    F    ^

-------------------------------------------------
 ACCESSED TOKEN OPEN_BRACKET: (
 Action: Shifting token value:: ( And going to state: 8 Printing automata_states_stack: 0    1    22    4    20    8
 Printing grammar_symbols_stack: $    E    +    F    ^    (

-------------------------------------------------

 ACCESSED TOKEN NUMBER: 9
 Action: Shifting token value:: 9 And going to state: 17 Printing automata_states_stack: 0    1    22    4    20    8    17
 Printing grammar_symbols_stack: $    E    +    F    ^    (    num

-------------------------------------------------

 ACCESSED TOKEN  PLUS: +
 Action: Reduce using production number: 10  that is the production:  B  <-  num
 Printing automata_states_stack: 0    1    22    4    20    8    11
 Printing grammar_symbols_stack: $    E    +    F    ^    (    B

-------------------------------------------------
 Action: Reduce using production number: 8  that is the production:  A  <-  B
 Printing automata_states_stack: 0    1    22    4    20    8    12
 Printing grammar_symbols_stack: $    E    +    F    ^    (    A

-------------------------------------------------
 Action: Reduce using production number: 6  that is the production:  F  <-  A
 Printing automata_states_stack: 0    1    22    4    20    8    13
 Printing grammar_symbols_stack: $    E    +    F    ^    (    F

-------------------------------------------------
 Action: Reduce using production number: 4  that is the production:  T  <-  F
 Printing automata_states_stack: 0    1    22    4    20    8    14
 Printing grammar_symbols_stack: $    E    +    F    ^    (    T

-------------------------------------------------
 Action: Reduce using production number: 2  that is the production:  E  <-  T
 Printing automata_states_stack: 0    1    22    4    20    8    10
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E

-------------------------------------------------
 Action: Shifting token value:: + And going to state: 30 Printing automata_states_stack: 0    1    22    4    20    8    10    30
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    +

-------------------------------------------------

 ACCESSED TOKEN UNMINUS: -
 Action: Shifting token value:: - And going to state: 15 Printing automata_states_stack: 0    1    22    4    20    8    10    30    15
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    +    -
```

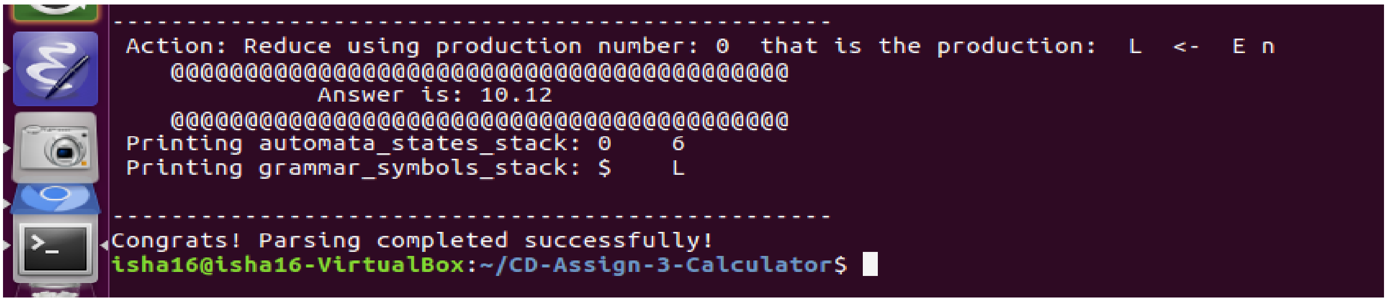


isha16@isha16-VirtualBox: ~/CD-Assign-3-Calculator

```
-------------------------------------------------

 ACCESSED TOKEN NUMBER: 9
 Action: Shifting token value:: 9 And going to state: 17 Printing automata_states_stack: 0    1    22    4    20    8    10    30    15    17
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    +    -    num

-------------------------------------------------
ACCESSED TOKEN CLOSE_BRACKET : )
 Action: Reduce using production number: 10  that is the production:  B  <-  num
 Printing automata_states_stack: 0    1    22    4    20    8    10    30    15    27
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    +    -    B

-------------------------------------------------
 Action: Reduce using production number: 7  that is the production:  A  <-  - B
 Printing automata_states_stack: 0    1    22    4    20    8    10    30    12
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    +    A

-------------------------------------------------
 Action: Reduce using production number: 6  that is the production:  F  <-  A
 Printing automata_states_stack: 0    1    22    4    20    8    10    30    13
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    +    F

-------------------------------------------------
 Action: Reduce using production number: 4  that is the production:  T  <-  F
 Printing automata_states_stack: 0    1    22    4    20    8    10    30    32
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    +    T

-------------------------------------------------
 Action: Reduce using production number: 1  that is the production:  E  <-  E + T
 Printing automata_states_stack: 0    1    22    4    20    8    10
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E

-------------------------------------------------
 Action: Shifting token value:: ) And going to state: 31 Printing automata_states_stack: 0    1    22    4    20    8    10    31
 Printing grammar_symbols_stack: $    E    +    F    ^    (    E    )

-------------------------------------------------

 ACCESSED TOKEN  PLUS: +
 Action: Reduce using production number: 9  that is the production:  B  <-  ( E )
 Printing automata_states_stack: 0    1    22    4    20    2
 Printing grammar_symbols_stack: $    E    +    F    ^    B

-------------------------------------------------
 Action: Reduce using production number: 8  that is the production:  A  <-  B
 Printing automata_states_stack: 0    1    22    4    20    24
 Printing grammar_symbols_stack: $    E    +    F    ^    A

-------------------------------------------------
 Action: Reduce using production number: 5  that is the production:  F  <-  F ^ A
 Printing automata_states_stack: 0    1    22    4
 Printing grammar_symbols_stack: $    E    +    F
```



```
--------------------------------------------------
 Action: Reduce using production number: 4  that is the production:  T  <-  F
 Printing automata_states_stack: 0    1    22    23
 Printing grammar_symbols_stack: $    E    +    T

--------------------------------------------------
 Action: Reduce using production number: 1  that is the production:  E  <-  E + T
 Printing automata_states_stack: 0    1
 Printing grammar_symbols_stack: $    E

--------------------------------------------------
 Action: Shifting token value:: + And going to state: 22 Printing automata_states_stack: 0    1    22
 Printing grammar_symbols_stack: $    E    +

--------------------------------------------------

 ACCESSED TOKEN NUMBER: 3.12
 Action: Shifting token value:: 3.12 And going to state: 9 Printing automata_states_stack: 0    1    22    9
 Printing grammar_symbols_stack: $    E    +    num

--------------------------------------------------
ACCESSED TOKEN ENTER :

 Action: Reduce using production number: 10  that is the production:  B  <-  num
 Printing automata_states_stack: 0    1    22    2
 Printing grammar_symbols_stack: $    E    +    B

--------------------------------------------------
 Action: Reduce using production number: 8  that is the production:  A  <-  B
 Printing automata_states_stack: 0    1    22    3
 Printing grammar_symbols_stack: $    E    +    A

--------------------------------------------------
 Action: Reduce using production number: 6  that is the production:  F  <-  A
 Printing automata_states_stack: 0    1    22    4
 Printing grammar_symbols_stack: $    E    +    F

--------------------------------------------------
 Action: Reduce using production number: 4  that is the production:  T  <-  F
 Printing automata_states_stack: 0    1    22    23
 Printing grammar_symbols_stack: $    E    +    T

--------------------------------------------------
 Action: Reduce using production number: 1  that is the production:  E  <-  E + T
 Printing automata_states_stack: 0    1
 Printing grammar_symbols_stack: $    E

--------------------------------------------------
 Action: Shifting token value:: n And going to state: 21 Printing automata_states_stack: 0    1    21
 Printing grammar_symbols_stack: $    E    n

--------------------------------------------------
```



```
--------------------------------------------------
 Action: Reduce using production number: 0  that is the production:  L  <-  E n
    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                Answer is: 10.12
    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
 Printing automata_states_stack: 0    6
 Printing grammar_symbols_stack: $    L

--------------------------------------------------
Congrats! Parsing completed successfully!
isha16@isha16-VirtualBox:~/CD-Assign-3-Calculator$ 
```