test1.txt

normal program input

```
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
[$, program_end, stmts, ;, declarators', exp, =]
[$, program_end, stmts, ;, declarators', exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', factor]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', number]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term']
[$, program_end, stmts, ;, declarators', exp', simple_exp']
[$, program_end, stmts, ;, declarators', exp']
[$, program_end, stmts, ;, declarators']
```

...

Output

```
[$, program_end, stmts, else_part, elseif_part, end, stmts, ;, ), disoption]
[$, program_end, stmts, else_part, elseif_part, end, stmts, ;, ), stringLiteral]
[$, program_end, stmts, else_part, elseif_part, end, stmts, ;, )]
[$, program_end, stmts, else_part, elseif_part, end, stmts, ;]
[$, program_end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, else_part, elseif_part, end]
[$, program_end, stmts, else_part, elseif_part]
[$, program_end, stmts, else_part]
[$, program_end, stmts]
[$, program_end]
[$]

>> Parsing OK
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 %
```

test2.txt

normal program input

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/test2.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
[$, program_end, stmts, ;, declarators', exp, =]
[$, program_end, stmts, ;, declarators', exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', factor]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', number]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term']
[$, program_end, stmts, ;, declarators', exp', simple_exp']
[$, program_end, stmts, ;, declarators', exp']
[$, program_end, stmts, ;, declarators']
```

• • •

Output

```
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, atomicStmt]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), disoption, (, display]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), disoption, (]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), disoption]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), stringLiteral]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, )]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, stmt]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, atomicStmt]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, break]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end]
[$, program_end, stmts, end, stmts, else_part, elseif_part]
[$, program_end, stmts, end, stmts, else_part]
[$, program_end, stmts, end, stmts]
[$, program_end, stmts, end]
[$, program_end, stmts]
[$, program_end]
[$]

>> Parsing OK
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 %
```

test3.txt

normal program input

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/test3.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
[$, program_end, stmts, ;, declarators', exp, =]
[$, program_end, stmts, ;, declarators', exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', factor]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', number]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term']
[$, program_end, stmts, ;, declarators', exp', simple_exp']
[$, program_end, stmts, ;, declarators', exp']
[$, program_end, stmts, ;, declarators']
[$, program_end, stmts, ;, declarators', option, id, ,]
```

• • •

Output

```
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), disoption, (, display]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), disoption, (]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), disoption]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, ), stringLiteral]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, )]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, stmt]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, atomicStmt]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, break]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end]
[$, program_end, stmts, end, stmts, else_part, elseif_part]
[$, program_end, stmts, end, stmts, else_part]
[$, program_end, stmts, end, stmts]
[$, program_end, stmts, end]
[$, program_end, stmts]
[$, program_end]
[$]

>> Parsing OK
```

test4.txt

```
program test4
   program_begin
       integer  Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour=3;
       for (integer i=0;  i <= Time_10_24+25 ; i+= i+1)
          begin
              display("test 4");
          end
       if (Time_10_24 < 10)
            begin
                display("the second test input");
                Hour_In_Day = 24;
            end
```

normal program input

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/test4.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
[$, program_end, stmts, ;, declarators', exp, =]
[$, program_end, stmts, ;, declarators', exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', factor]
```

...

Output

```
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, stmt]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, atomicStmt]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;, break]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts, ;]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, end, stmts, else_part, elseif_part, end]
[$, program_end, stmts, end, stmts, else_part, elseif_part]
[$, program_end, stmts, end, stmts, else_part]
[$, program_end, stmts, end, stmts]
[$, program_end, stmts, end]
[$, program_end, stmts]
[$, program_end]
[$]

>> Parsing OK
```

testError1.txt

LL1ParserInputs > testError1.txt

```
1   program1 Error_input1
2       program_begin
3               integer  Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour;
4               if (Time_10_24 < 10)
5                 begin
6                     display("error input 1");
7                     Hour_In_Day = 24;
8                 end
9             elseif (Time_10_24 <= 20)
10                begin
11                    Minute_In_Hour = Hour_In_Day * 60;
12                    display("Good Day. Peace_to_you.");
13                end
14      program_end
15
```

Output

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/testError1.txt
[$, start]

error: [program] keyword not matched
>> Parsing Failed
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 %
```

testError2.txt

```
program Error_input2
   program_begin
            integer  Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour;
            if (Time_10_24 < 10)
              begin
                 display("error input 2");
                 Hour_In_Day = 24;
              end
          elseif (Time_10_24 <= 20)
              begin
                 Minute_In_Hour = Hour_In_Day * 60;
                 display("Good Day. Peace_to_you.");
              end
 program9_end
```

Output

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/testError2.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
```

...

```
[$, program_end, stmts, else_part, elseif_part, end, stmts, ;]
[$, program_end, stmts, else_part, elseif_part, end, stmts]
[$, program_end, stmts, else_part, elseif_part, end]
[$, program_end, stmts, else_part, elseif_part]
[$, program_end, stmts, else_part]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, assignStmt]
[$, program_end, stmts, ;, exp, op, id]
[$, program_end, stmts, ;, exp, op]

error: [program_end] keyword not matched
>> Parsing Failed
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 %
```

testError3.txt

```
program Error_input3
    program_begin
            for  Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour;
              if (Time_10_24 < 10)
                begin
                    display("error input 3");
                    Hour_In_Day = 24;
                end
           elseif (Time_10_24 <= 20)
                begin
                    Minute_In_Hour = Hour_In_Day * 60;
                    display("Good Day. Peace_to_you.");
                end
   program_end
```

Output

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/testError3.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, forStmt]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, for_cond, ;, exp, =, id, integer, (, for]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, for_cond, ;, exp, =, id, integer, (]

declaration keyword not matched ..
>> Parsing Failed
```

testError4.txt

```
program Error_input4
    program_begin
            Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour;
            if (Time_10_24 < 10)
              begin
                display("error input 4");
                Hour_In_Day = 24;
              end
          elseif (Time_10_24 <= 20)
              begin
                Minute_In_Hour = Hour_In_Da
                display("Good Day. Peace_t
              end
    program_end
```

Output

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/testError4.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, assignStmt]
[$, program_end, stmts, ;, exp, op, id]
[$, program_end, stmts, ;, exp, op]
[$, program_end, stmts, ;, exp, =]
[$, program_end, stmts, ;, exp]
[$, program_end, stmts, ;, exp', simple_exp]
[$, program_end, stmts, ;, exp', simple_exp', term]
[$, program_end, stmts, ;, exp', simple_exp', term', factor]
[$, program_end, stmts, ;, exp', simple_exp', term', number]
[$, program_end, stmts, ;, exp', simple_exp', term']
[$, program_end, stmts, ;, exp', simple_exp']
[$, program_end, stmts, ;, exp']
[$, program_end, stmts, ;]

; is expected but , is given
maybe declaration keyword is missing ..
```

testError5.txt

```
program Error_input5
    program_begin
            integer  Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour;
            if (Time_10_24 < 10)
                display("error input 5");
                Hour_In_Day = 24;
            end
        elseif (Time_10_24 <= 20)
            begin
                Minute_In_Hour = Hour_In_Day * 60;
                display("Good Day. Peace_to_you.");
            end
    program_end
```

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/testError5.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
[$, program_end, stmts, ;, declarators', exp, =]
[$, program_end, stmts, ;, declarators', exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', factor]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term', number]
[$, program_end, stmts, ;, declarators', exp', simple_exp', term']
```

...

Output

```
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp']
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp']
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp, compop]
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp, <]
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp]
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp', term]
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp', term', factor]
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp', term', number]
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp', term']
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp', simple_exp']
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, ), exp']
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin, )]
[$, program_end, stmts, else_part, elseif_part, end, stmts, begin]

begin keyword is missing in ifStmt
>> Parsing Failed
```

testError6.txt

```
program Error_input6
   program_begin
          integer  Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour=3;
          for (integer i=0;  i; i++)
             begin
                 display("error input 6");
             end
          if (Time_10_24 < 10)
```

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/testError6.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
[$, program_end, stmts, ;, declarators', exp, =]
```

...

Output

```
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, for_cond, ;, exp', simple_exp', term']
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, for_cond, ;, exp', simple_exp']
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, for_cond, ;, exp']
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, for_cond, ;]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, for_cond]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp', term]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp', term', fac
tor]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp', term', id]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp', term']
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp']
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop]

error: syntax error in test part of for loop
>> Parsing Failed
```

testError7.txt

```
program Error_input7
   program_begin
          integer  Time_10_24 = 2206, Hour_In_Day = 0, Minute_In_Hour=3;
          for (integer i=0;  i == "errr" ; i++)
             begin
                 display("error input 7");
             end
```

```
(base) hanseunghwa@hanseunghwaui-MacBookPro hw3 % java LLparser LL1ParserInputs/testError7.txt
[$, start]
[$, program_end, stmts, program_begin, id, program]
[$, program_end, stmts, program_begin, id]
[$, program_end, stmts, program_begin]
[$, program_end, stmts]
[$, program_end, stmts, stmt]
[$, program_end, stmts, defineStmt]
[$, program_end, stmts, ;, declarators, integer]
[$, program_end, stmts, ;, declarators]
[$, program_end, stmts, ;, declarators', option, id]
[$, program_end, stmts, ;, declarators', option]
```

...

Output

```
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp', term', fac
tor]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp', term', id]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp', term']
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop, simple_exp']
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, compop]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp, ==]
[$, program_end, stmts, end, stmts, begin, ), for_update, ;, simple_exp]

error: not number in test part of for loop
>> Parsing Failed
```