Test Code:	Lexer/Parser Output:

{print ("start")
int a
int b
a = 1
b = true
c = 2 + 4
{
boolean c
}
print ("end")
}$
	Compilation Started
	Lex returned
	[{,print,(,"start",),int,a,int,b,a,=,1,b,=,true,c,=,2,+,4,{,boolean,c,},print,(,"end",),},$]
	Lex Errors: 0 | Lex Warnings: 0
	Token Accepted: Expecting token of value {, found token of value {
	Token Accepted: Expecting token of value print, found token of value print
	Token Accepted: Expecting token of value (, found token of value (
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value "start"
	Token Accepted: Expecting token of value), found token of value)
	Token Accepted: Expecting token of value int, string or boolean, found token of value int
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value a
	Token Accepted: Expecting token of value int, string or boolean, found token of value int
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value b
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value a
	Token Accepted: Expecting token of value =, found token of value =
	Token Accepted: Expecting token of value one digit, found token of value 1
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value b
	Token Accepted: Expecting token of value =, found token of value =
	Token Accepted: Expecting token of value true or false, found token of value true
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value c
	Token Accepted: Expecting token of value =, found token of value =
	Token Accepted: Expecting token of value one digit, found token of value 2
	Token Accepted: Expecting token of value +, found token of value +
	Token Accepted: Expecting token of value one digit, found token of value 4
	Token Accepted: Expecting token of value {, found token of value {
	Token Accepted: Expecting token of value int, string or boolean, found token of value boolean
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value c
	Token Accepted: Expecting token of value }, found token of value }
	Token Accepted: Expecting token of value print, found token of value print
	Token Accepted: Expecting token of value (, found token of value (
	Token Accepted: Expecting token of value one lowercase char a-z, found token of value "end"
	Token Accepted: Expecting token of value), found token of value)
	Token Accepted: Expecting token of value }, found token of value }

	Token Accepted: Expecting token of value $, found token of value $ Parse Errors: 0 | Parse Warnings: 0
{{{int a}}	Compilation Started Lex returned [{,{,{,int,a,},},$] Warning: EOF symbol not found at end of program, $ inserted at end Lex Errors: 0 | Lex Warnings: 1 Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value int, string or boolean, found token of value int Token Accepted: Expecting token of value one lowercase char a-z, found token of value a Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value }, found token of value } Parse Error: Line 1, Found $, Expecting token of value } Token Accepted: Expecting token of value $, found token of value $ Parse Errors: 1 | Parse Warnings: 0
{a = 2 + 22}$	Compilation Started Lex returned [{,a,=,2,+,2,2,},$] Lex Errors: 0 | Lex Warnings: 0 Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value one lowercase char a-z, found token of value a Token Accepted: Expecting token of value =, found token of value = Token Accepted: Expecting token of value one digit, found token of value 2 Token Accepted: Expecting token of value +, found token of value + Token Accepted: Expecting token of value one digit, found token of value 2 Parse Error: Line 1, Found 2, Expecting token of value } Parse Error: Line 1, Found }, Expecting token of value $ Parse Errors: 2 | Parse Warnings: 0

{
{{print((true != false))}}
while true {
string z
z = "z"
string y
y = "y"
$
}
}$
	Compilation Started
	Lex returned
	[{,{,{,print,(,(,true,!=,false,),),},},while,true,{,string,z,z,=,"z",string,y,y,=,"y",$]
	Warning: EOF symbol found in middle of program, code afterward ignored
	Lex Errors: 0 | Lex Warnings: 1
	Token Accepted: Expecting token of value {, found token of value {
	Token Accepted: Expecting token of value {, found token of value {
	Token Accepted: Expecting token of value {, found token of value {
	Token Accepted: Expecting token of value print, found token of value print
	Token Accepted: Expecting token of value (, found token of value (
	Token Accepted: Expecting token of value (, found token of value (
	Token Accepted: Expecting token of value true or false, found token of value true
	Token Accepted: Expecting token of value !=, found token of value !=
	Token Accepted: Expecting token of value true or false, found token of value false
	Token Accepted: Expecting token of value), found token of value)

	Token Accepted: Expecting token of value), found token of value) Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value while, found token of value while Token Accepted: Expecting token of value true or false, found token of value true Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value int, string or boolean, found token of value string Token Accepted: Expecting token of value one lowercase char a-z, found token of value z Token Accepted: Expecting token of value one lowercase char a-z, found token of value z Token Accepted: Expecting token of value =, found token of value = Token Accepted: Expecting token of value one lowercase char a-z, found token of value "z" Token Accepted: Expecting token of value int, string or boolean, found token of value string Token Accepted: Expecting token of value one lowercase char a-z, found token of value y Token Accepted: Expecting token of value one lowercase char a-z, found token of value y Token Accepted: Expecting token of value =, found token of value = Token Accepted: Expecting token of value one lowercase char a-z, found token of value "y" Parse Error: Line 8, Found $, Expecting token of value } Parse Error: Line 8, Found $, Expecting token of value } Token Accepted: Expecting token of value $, found token of value $ Parse Errors: 2 | Parse Warnings: 0
{int int int int int int int int int}$	Compilation Started Lex returned [{,int,int,int,int,int,int,int,int,int,},$] Lex Errors: 0 | Lex Warnings: 0 Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value int, string or boolean, found token of value int Parse Error: Line 1, Found int, Expecting token of value one lowercase char a-z Token Accepted: Expecting token of value int, string or boolean, found token of value int Parse Error: Line 1, Found int, Expecting token of value one lowercase char a-z Token Accepted: Expecting token of value int, string or boolean, found token of value int Parse Error: Line 1, Found int, Expecting token of value one lowercase char a-z Token Accepted: Expecting token of value int, string or boolean, found token of value int Parse Error: Line 1, Found int, Expecting token of value one lowercase char

	a-z Token Accepted: Expecting token of value int, string or boolean, found token of value int Parse Error: Line 1, Found int, Expecting token of value one lowercase char a-z Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value $, found token of value $ Parse Errors: 5 | Parse Warnings: 0
{{{{{inta}intb}intc}intd}inte}$	Compilation Started Lex returned [{,{,{,{,{,int,a,},int,b,},int,c,},int,d,},int,e,},$] Lex Errors: 0 | Lex Warnings: 0 Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value {, found token of value { Token Accepted: Expecting token of value int, string or boolean, found token of value int Token Accepted: Expecting token of value one lowercase char a-z, found token of value a Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value int, string or boolean, found token of value int Token Accepted: Expecting token of value one lowercase char a-z, found token of value b Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value int, string or boolean, found token of value int Token Accepted: Expecting token of value one lowercase char a-z, found token of value c Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value int, string or boolean, found token of value int Token Accepted: Expecting token of value one lowercase char a-z, found token of value d Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value int, string or boolean, found token of value int Token Accepted: Expecting token of value one lowercase char a-z, found token of value e Token Accepted: Expecting token of value }, found token of value } Token Accepted: Expecting token of value $, found token of value $ Parse Errors: 0 | Parse Warnings: 0