termvar, x, y
funcname, `name`
indecies, i, j

types, T, A, B, C	::=		Types
	|	Bool	
Γ	::=		Typing Context
	|	$\emptyset$	
	|	$x_1 : T_1, \ldots, x_i : T_i$	
program, p	::=		Programs
	|	func name$(x_1 : T_1, \ldots, x_i : T_i) \to T\{body\}$	
	|	$p_1\ p_2$	
	|	(p)	
body	::=		Function Bodies
	|	$asgn; t$	Body with assignments
	|	t	Body without assignments
lv	::=		Assignment Tags
	|	let	Use x at least once
	|	var	Use x any number of times
assignemnts, *asgn*	::=		Variable Assignments
	|	$lv_1\ x_1 : T_1 = b_1; \ldots; lv_j\ x_j : T_j = b_j$	
t	::=		Terms
	|	x	A variable
	|	T	Logical true
	|	F	Logical false
	|	if b_1 then t_1 else t_2	Pattern Matching for booleans
	|	name$(b_1, \ldots, b_i)$	Function application
	|	return b	Return of a term
	|	$EC[t]$	Plugging the hole in EC gives
	|	(t)	S
b	::=		Basic Terms
	|	x	A variable
	|	T	Logical true
	|	F	Logical false
	|	if b_1 then t_2 else t_3	Pattern Matching for booleans
	|	name$(b_1, \ldots, b_i)$	Function application
	|	$EC[b]$	Plugging the hole in EC gives
	|	(b)	S
v	::=		Values
	|	T	
	|	F	
Δ	::=		Contexts of Function Definitions
	|	p_1	
	|	Δ_1, Δ_2	

TP	::=		List of Term Parameters
	|	$\emptyset$	Empty List
	|	b, TP	Term Argument
CP	::=		Evaluation Contexts for Parameters
	|	$\emptyset$	Empty List
	|	EC, TP	Context Evaluation Argument
	|	b, CP_2	Term Argument
$evalctx,\ EC$	::=		Evaluation Contexts
	|	$\square$	The hole (location of the evaluation point)
	|	T	
	|	F	
	|	$\mathsf{if}\ EC\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3$	
	|	$\mathtt{name}(CP)$	
	|	$\mathsf{return}\ b$	
	|	(EC)	S

$\boxed{\Delta; \Gamma \vdash t : T}$ Type Checking

$$\frac{x : T \in \Gamma}{\Delta; \Gamma \vdash x : T} \quad \text{T\_Var}$$

$$\frac{}{\Delta; \Gamma \vdash T : \mathsf{Bool}} \quad \text{T\_True}$$

$$\frac{}{\Delta; \Gamma \vdash \mathsf{F} : \mathsf{Bool}} \quad \text{T\_False}$$

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash t_1 : T \\ \Delta; \Gamma \vdash t_2 : T \quad \Delta; \Gamma \vdash b_1 : \mathsf{Bool}\end{array}}{\Delta; \Gamma \vdash \mathsf{if}\ b_1\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : T} \quad \text{T\_If}$$

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash b_1' : T_1, \ldots, \Delta; \Gamma \vdash b_i' : T_i \\ \Delta; \Gamma, x_1 : T_1, \ldots, x_i : T_i \vdash b_1 : A_1 \\ \Delta; \Gamma, x_1 : T_1, \ldots, x_i : T_i, y_1 : A_1, \ldots, y_{j-1} : A_{j-1} \vdash b_j : A_j \\ \Delta; \Gamma, x_1 : T_1, \ldots, x_i : T_i, y_1 : A_1, \ldots, y_j : A_j \vdash t : T \\ (\mathsf{func}\,\mathtt{name}(x_1 : T_1, \ldots, x_i : T_i) \to T\{lv_1\ y_1 : A_1 = b_1; \ldots; lv_j\ y_j : A_j = b_j; t\}) \in \Delta\end{array}}{\Delta; \Gamma \vdash \mathtt{name}(b_1', \ldots, b_i') : T} \quad \text{T\_App}$$

$$\frac{\Delta; \Gamma \vdash b : T}{\Delta; \Gamma \vdash \mathsf{return}\ b : T} \quad \text{T\_Return}$$

$\boxed{\Delta \vdash t_1 \rightsquigarrow t_2}$ Single-Step Reduction for Terms

$$\frac{}{\Delta \vdash \mathsf{if}\ T\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \rightsquigarrow t_1} \quad \text{IfT}$$

$$\frac{}{\Delta \vdash \mathsf{if}\ \mathsf{F}\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \rightsquigarrow t_2} \quad \text{IfF}$$

$$\frac{}{\Delta \vdash \mathsf{return}\ b \rightsquigarrow b} \quad \text{Return}$$

$$\frac{(\mathsf{func}\,\mathtt{name}(x_1 : T_1, \ldots, x_i : T_i) \to T\{lv_1\ y_1 : A_1 = b_1; \ldots; lv_j\ y_j : A_j = b_j; t\}) \in \Delta}{\Delta \vdash \mathtt{name}(v_1, \ldots, v_i) \rightsquigarrow [v_1, \ldots, v_i/x_1, \ldots, x_i][b_1, \ldots, b_j/y_1, \ldots, y_j]t} \quad \text{Beta}$$

$\boxed{\Delta \vdash t_1 \rightsquigarrow^* t_2}$ Multi-Step Reduction for Terms

$$\frac{\Delta \vdash t_1 \rightsquigarrow t_2}{\Delta \vdash EC[t_1] \rightsquigarrow^* EC[t_2]} \textsc{Step}$$

$$\frac{\Delta \vdash t_1 \rightsquigarrow t_2 \quad \Delta \vdash EC[t_2] \rightsquigarrow^* EC[t_3]}{\Delta \vdash EC[t_1] \rightsquigarrow^* EC[t_3]} \textsc{Mult}$$

```
Definition rules:         12 good    0 bad
Definition rule clauses: 24 good    0 bad
```