Formula					Variables
precondition	statement		postcondition {this.floors.length = numFloors & (\forall int k;		
{numFloors >= 0}	state	statement		is.floors.length -> .thisFloorID = k & k].env = this))))}	PARAM int numFloors  Global Conditions
				,	numFloors >= 0
Composition					index >= 0
precondition		postcondition			•
{numFloors >= 0}		{this.floors.length = numFloors & (\forall int k; ((k>0 & k <this.floors.length -=""> (this.floors[k].thisFloorID = k &amp; this.floors[k].env = this))))}</this.floors.length>		thisFloorID = k &	
statement 1 intermediate		te condition	stat	ement 2	
statement1 & this.floors. <cre< td=""><td>ex = 0 reated&gt; = TRUE gth = numFloors}</td><td colspan="2">statement2</td><td></td></cre<>		ex = 0 reated> = TRUE gth = numFloors}	statement2		
Statement1 $\sqrt{}$	ement1 $$ RepetitionStatement1 $$				
	ement postcondition   Repetition Statement DOOD				~
& and a second s		numFloors &(\forall int k;(((k>0 & k <index &="" -="" k<this.floors.length)=""> (this.floors[k].thisFloorID = k &amp;</index>		·d	variant
{numFloors >= 0} index = 0; this.floors. ed> = T	numFloors & k <index &="" (this.floors<="" k="" length="" td=""><td>oors.length (</td><td>(this.floors.length - index)</td></index>			oors.length (	(this.floors.length - index)
		precondition loop state		ement	postcondition
n k-		{(index >= 0 & this.floors.length = numFloors &(\forall int k;(((k>0 & k <index &="" -="" k<this.floors.length)=""> (this.floors[k].thisFloorID = k &amp; this.floors[k].env = this ))))) &amp; (index &lt; this.floors.length)}</index>		num k <in (th</in 	ex >= 0 & this.floors.length = Floors &(\forall int k;(((k>0 & dex & k <this.floors.length) -=""> is.floors[k].thisFloorID = k &amp; his.floors[k].env = this ))))}</this.floors.length)>
Statement2					
precondition		statement		postcondition	
{(index >= 0 & this.floors.length = numFloors int k;(((k>0 & k <index &="" k<this.floors.leng<br="">(this.floors[k].thisFloorID = k &amp; this.floors[k this ))))) &amp; (index &lt; this.floors.length</index>	th) -> this.floors[ing].env =	this.floors[index] = new Floor(this, index); index = index + 1;		{index >= 0 & this.floors.length = numFloors &(\forall int k;(((k>0 & k <index &="" -="" k<this.floors.length)=""> (this.floors[k].thisFloorID = k &amp; this.floors[k].env = this ))))}</index>	