program	::=	dec-list
exp	::=	Boolean-exp
		| **let** dec-list **in** exp
		| **if** (Boolean-exp) **then** (exp) **else** (exp)
atomic-exp	::=	integer | (exp) | identifier {parameter-exp}
parameter-exp	::=	integer | (exp) | identifier
dec-list	::=	dec { dec }
dec	::=	identifier = exp | identifier argument-list = exp
argument-list	::=	identifier {identifier}
Boolean-exp	::=	relational-exp {Boolean-operator relational-exp}
Boolean-operator	::=	&& | ||
relational-exp	::=	list-exp [relational-operator list-exp]
relational-operator	::=	== | / = | < | <= | > | >=
list-exp	::=	primary-exp : list-exp | **tail** atomic-exp | [ ] | primary-exp
primary-exp	::=	term {adding-operator term}
adding-operator	::=	+ | −
term	::=	factor {multiplying-operator factor}
multiplying-operator	::=	* | /
factor	::=	[**head**] atomic-exp