

FIRST	FOLLOW	Nonterminal	begin	end	pass
{begin, pass, id, if, while}	{\$, begin, pass, id, if, while, end}	Prog	Prog -> begin StmtSeq end		Prog -> Stmt
{pass, id, if, while}	{end}	StmtSeq			StmtSeq -> Stmt StmtTail
{', pass, id, if, while}	{end}	StmtTail		StmtTail -> ''	StmtTail -> Stmt StmtTail
{pass, id, if, while}	{\$, end, pass, id, if, while, begin}	Stmt			Stmt -> pass
{id}	{\$, end, pass, id, if, while, begin}	Assgn			
{if}	{\$, end, pass, id, if, while, begin}	If			
{while}	{\$, end, pass, id, if, while, begin}	While			
{id, literal, op}	{\$, end, pass, id, if, while, begin, literal, op}	Expr			

id	:=	if	while	literal	op	\$
Prog -> Stmt		Prog -> Stmt	Prog -> Stmt			
StmtSeq -> Stmt StmtTail		StmtSeq -> Stmt StmtTail	StmtSeq -> Stmt StmtTail			
StmtTail -> Stmt StmtTail		StmtTail -> Stmt StmtTail	StmtTail -> Stmt StmtTail			
Stmt -> Assgn		Stmt -> If	Stmt -> While			
Assgn -> id := Expr						
		If -> if Expr Prog Prog				
			While -> while Expr Prog			
Expr -> id				Expr -> literal	Expr -> op Expr Expr	