3 4	S'	S	S'.IR = S.IR print(S'.IR)	
	S	<body></body>	S.IR = <body>.IR</body>	
	S <body></body>	<body> S  <decl></decl></body>	S.IR = <body>.IR + S.IR <b>.IR = <d>.IR</d></b></body>	
5	<decl></decl>	<vardecl></vardecl>	if global declaration is not a constant: ERROR <d>.IR = <vd>.IR</vd></d>	Trans
6	<decl></decl>	<funcdecl></funcdecl>	<d>.IR = <fd>.IR  ID exists: ERROR</fd></d>	
7	<vardecl></vardecl>	int ID ;	ID.valType = 'int' <vd>.IR = 'ID = 0'  insert into var symbol table</vd>	
8	<vardecl></vardecl>	int ID = <exprsn> ;</exprsn>	ID exists: ERROR ID.valType = 'int' if <e>.valType is not 'int': ERROR <vd>.IR = <e>.IR + 'ID = <e>.val' insert into var symbol table</e></e></vd></e>	
9	<vardecl></vardecl>	float ID ;	ID exists: ERROR ID.valType = 'float' <vd>.IR = 'ID = 0e+00' insert into var symbol table</vd>	
10	<vardecl></vardecl>	float ID = <exprsn> ;</exprsn>	ID exists: ERROR ID.valType = 'float' if <e>.valType is not 'float': ERROR <vd>.IR = <e>.IR + 'ID = <e>.val' insert into var symbol table</e></e></vd></e>	
			ID.val exists: ERROR <fd>.returnType != <sb>.rT: ERROR</sb></fd>	
11	<funcdecl></funcdecl>	int ID( <formalparams>)<stmtblock></stmtblock></formalparams>	<fd>.IR = 'def func(' + <fp>.IR + ')' + <sb>.IR + '}' <fd>.returnType = 'int' <fd>.paramName = <fp>.paramName <fd>.paramType = <fp>.paramType insert into func symbol table pop out the params from var table</fp></fd></fp></fd></fd></sb></fp></fd>	Trans
			ID.val exists: ERROR	
12	<funcdecl></funcdecl>	float ID ( <formalparams>) <stmtblock></stmtblock></formalparams>	<pre><fd>.returnType != <sb>.rT: ERROR <fd>.IR = 'def func(' + <fp>.IR + ')' + <sb>.IR + '}'</sb></fp></fd></sb></fd></pre>	Trans
			ID.val exists: ERROR	
13	<funcdecl></funcdecl>	void ID( <formalparams>)<stmtblock></stmtblock></formalparams>	<pre><fd>.returnType != <sb>.rT: ERROR <fd>.IR = 'def func(' + <fp>.IR + ') {' + <sb>.IR + 'ret void }'</sb></fp></fd></sb></fd></pre>	Trans
14	<formalparams></formalparams>	<paramlist></paramlist>	<pre><fp>.paramType = <pl>.paramType <fp>.paramName = <pl>.paramName</pl></fp></pl></fp></pre>	
14			<pre><fp>.IR = <pl>.IR All params push into var table </pl></fp></pre> <pre><fp>.paramType = []</fp></pre>	
15	<formalparams> <formalparams></formalparams></formalparams>	void	<pre><fp>.paramName = [] </fp></pre> <pre><fp>.paramType = []</fp></pre>	
17	<pre><formalparams> </formalparams></pre> <pre><paramlist></paramlist></pre>	ε <param/>	<pre><fp>.paramName = []  <pl>.paramType.putHead(<p>.type) <pl>.paramName.putHead(<p>.name)</p></pl></p></pl></fp></pre>	
			<pl>.IR = <p>.IR  <pl1>.paramType = <pl2>.paramType <pl1>.paramName = <pl2>.paramName</pl2></pl1></pl2></pl1></p></pl>	
18	<paramlist></paramlist>	<param/> , <paramlist></paramlist>	<pl1>.paramType.putHead(<p>.type) <pl1>.paramName.putHead(<p>.name) <pl1>.IR = <p>.IR + ', ' + <pl2>.IR</pl2></p></pl1></p></pl1></p></pl1>	
19	<param/>	int ID	ID.val is the same as a global var: ERROR <p>.type = 'int'  <p>.name = ID.val  <p>.IR = 'int %ID'</p></p></p>	
20	<param/>	float ID	ID.val is the same as a global var: ERROR <p>.type = 'float'  <p>.name = ID.val  <p>.IR = 'float %ID'</p></p></p>	
21	<stmtblock></stmtblock>	{ <stmts> }</stmts>	<sb>.IR = <s>.IR  <sb>.returnType = <s>.returnType  pop <s>.innerVarAmount vars from var symbol table</s></s></sb></s></sb>	
22	<stmts></stmts>	<stmt> <stmts></stmts></stmt>	<stmts1>.IR = <stmt>.IR + <stmts2>.IR  if <s> <stmts2> rT equal: <stmts1>.rT = <stmt>.rT  else: ERROR  <stmts1>.innerVarAmount = <stmt>.iVA + <stmts2>.iVA</stmts2></stmt></stmts1></stmt></stmts1></stmts2></s></stmts2></stmt></stmts1>	
23	<stmts></stmts>	<stmt></stmt>	<stmts1>.innerVarAmount = <stmt>.iVA + <stmts2>.iVA  <stmts>.IR = <s>.IR  <stmts>.returnType = <s>.returnType  <stmts>.innerVarAmount = <stmt>.innerVarAmount</stmt></stmts></s></stmts></s></stmts></stmts2></stmt></stmts1>	
24	<stmt></stmt>	<vardecl></vardecl>	<stmts>.innerVarAmount = <stmt>.innerVarAmount  <s>.IR = <vd>.IR  <s>.returnType = 'void'  <stmt>.innerVarAmount += 1 (default is 0)</stmt></s></vd></s></stmt></stmts>	
25	<stmt></stmt>	<lfstmt></lfstmt>	<s>.IR = <is>.IR <s>.returnType = <is>.returnType</is></s></is></s>	
26	<stmt></stmt>	<whilestmt></whilestmt>	<s>.IR = <ws>.IR <s>.returnType = <ws>.returnType</ws></s></ws></s>	
27	<stmt></stmt>	<returnstmt></returnstmt>	<s>.IR = <rs>.IR <s>.returnType = <rs>.returnType</rs></s></rs></s>	
28	<stmt></stmt>	<assignstmt></assignstmt>	<s>.IR = <as>.IR <s>.returnType = 'void'</s></as></s>	
29	<stmt></stmt>	ID <funccall> ;</funccall>	ID not in func table: ERROR <fc>.args does not match func table item: ERROR  if type of args don't match: ERROR  ID.funcReturnType is not void: ERROR</fc>	
			<s>.IR = <fc>.IR + 'call ID()'</fc></s>	
30	<assignstmt></assignstmt>	ID = <exprsn> ;</exprsn>	if ID not in var table: ERROR if ID and <e> valType not match: ERROR <as>.IR = <e>.IR + 'ID = <e>.val' if ID is global!</e></e></as></e>	Trar
31	<returnstmt></returnstmt>	return <exprsn> ;</exprsn>	<rs>.IR = <e>.IR + 'ret <e>.val ;' <rs>.returnType = <e>.varType</e></rs></e></e></rs>	
32	<returnstmt></returnstmt>	return ;	<rs>.IR = 'ret void' <rs>.returnType = 'void'</rs></rs>	
33	<whilestmt></whilestmt>	while ( <exprsn> ) <stmtblock></stmtblock></exprsn>	if <e>.valType is not 'int': ERROR <ws>.returnType = <sb>.returnType <ws>.IR = <e>.IR + 'goto L1' + 'L1:' + 'if(<e>.val == 1) goto L2 else L3:' + 'L2' + <sb>.IR + 'goto L1' + 'L3:'</sb></e></e></ws></sb></ws></e>	
	WOLL	if ( Farmer ) ObertPlants also ObertPlants	if <e>.valType is not 'int': ERROR <sb1>.rT equals to <sb2>.rT: <is>.rT = <sb1>.rT else: ERROR</sb1></is></sb2></sb1></e>	
34	<lfstmt></lfstmt>	if ( <exprsn> ) <stmtblock> else <stmtblock></stmtblock></stmtblock></exprsn>	< S>.IR = <e>.IR + 'if (<e>.val != 1) goto L1 else L2'</e></e>	
35	<ifstmt></ifstmt>	if ( <exprsn> ) <stmtblock></stmtblock></exprsn>	<pre></pre> <pre></pre> <pre><is>.returnType = <sb>.returnType <is>.IR = <e>.IR + 'if (<e>.val != 1) goto L1 else L2:'</e></e></is></sb></is></pre>	
36	<exprsn></exprsn>	<addexprsn></addexprsn>	<e>.val = <a>.val <e>.valType = <a>.valType <e>.IR = <a>.IR</a></e></a></e></a></e>	
			<a> <e1> type not match: ERROR   <e1>.IR = <a>.IR + <e1>.IR +   'newTemp1 = (<a>.val &lt; <e2>.val);   if newTemp1 goto L1 else L2</e2></a></e1></a></e1></e1></a>	
37	<exprsn></exprsn>	<addexprsn> &lt; <exprsn></exprsn></addexprsn>	L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int'	
			<e1>.val = newTemp <e1>.valType = newTemp.valType</e1></e1>	
			<a> <e1> type not match: ERROR   <e1>.IR = <a>.IR + <e1>.IR +   'newTemp1 = (<a>.val &lt;= <e2>.val);   if newTemp1 goto L1 else L2</e2></a></e1></a></e1></e1></a>	
38	<exprsn></exprsn>	<addexprsn> &lt;= <exprsn></exprsn></addexprsn>	L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int'	
			<e1>.val = newTemp <e1>.valType = newTemp.valType</e1></e1>	
			<a> <e1> type not match: ERROR  <e1>.IR = <a>.IR + <e1>.IR +  'newTemp1 = (<a>.val &gt; <e2>.val);  if newTemp1 goto L1 else L2</e2></a></e1></a></e1></e1></a>	
39	<exprsn></exprsn>	<addexprsn> &gt; <exprsn></exprsn></addexprsn>	I.4 T 0 4	
39		<addexprsn> &gt; <exprsn></exprsn></addexprsn>	L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int'	
39		<addexprsn> &gt; <exprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0	
39		<addexprsn> &gt; <exprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val);</e2></a></e1></a></e1></e1></a></e1></e1>	
39	<exprsn></exprsn>	<addexprsn> &gt; <exprsn> <addexprsn> &gt;= <exprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int'</e2></a></e1></a></e1></e1></a></e1></e1>	
	<exprsn></exprsn>		goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3</e2></a></e1></a></e1></e1></a></e1></e1>	
	<exprsn></exprsn>		goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR + 'newTemp1 = (<a>.val == <e2>.val); if newTemp1 goto L1 else L2</e2></a></e1></e1></a></e1></e1></e1></e2></a></e1></a></e1></e1></a></e1></e1>	
	<exprsn></exprsn>		goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val == <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int'</e2></a></e1></a></e1></e1></a></e1></e1></e2></a></e1></a></e1></e1></a></e1></e1>	
40		<addexprsn> &gt;= <exprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3:'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.valType = newTemp.valType  <a> <e1> inewTemp1 goto L1 else L2 L1: newTemp1 = (<a>.val == <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3</e2></a></e1></a></e1></e1></a></e1></e1></e2></a></e1></a></e1></e1></a></e1></e1>	
40		<addexprsn> &gt;= <exprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a><e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: 'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a><e1>.valType = newTemp.valType  <a><e1>.valType = newTemp.valType  L1: newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: 'newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: 'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a><e1>.val = newTemp <e1>.valType = newTemp.valType</e1></e1></a></e1></e1></e1></a></e1></a></e1></e1></e2></a></e1></a></e1></e1></a></e1></e1>	
40		<addexprsn> &gt;= <exprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>val = newTemp <e1>valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val &gt;= <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val == <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L3: 'newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val == <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L3: 'newTemp1 goto L1 else L2 L1: newTemp1 = (<a>.val == <e2>.val); if newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: 'newTemp2 = 0</e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e2></a></e1></a></e1></e1></a></e1></e1></e2></a></e1></a></e1></e1></a></e1></e1></e2></a></e1></a></e1></e1></a></e1></e1>	
41	<exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>.val = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val) = <e2>.val); if newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.IR = <a>.IR + <e1>.IR + 'newTemp1 = (<a>.val = <e2>.val); if newTemp1 goto L1 else L2 L1: newTemp1 = (<a>.val = <e2>.val); if newTemp1 = (<a>.val = <e2>.val); if newTemp2 = 0 goto L3 L2: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>.valType = newTemp <e1>.valType = newTemp.valType  <a> <e1> type not match: ERROR <e1>.val = newTemp <e1>.valType = newTemp <e1> .valType = newTemp.valType  L1: newTemp1 = (<a>.val = <e2>.val); if newTemp2 = 1 goto L3 L2: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3</e2></a></e2></a></e2></a></e2></a></e1></e1></e1></e1></a></e1></e1></e2></a></e2></a></e2></a></e1></a></e1></e1></a></e1></e2></a></e1></a></e1></e1></a></e1></e1>	
41	<exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn></exprsn></addexprsn></exprsn></addexprsn>	Goto L3   L2: newTemp2 = 0   goto L3   L3:newTemp.valType = 'Int'   <e1>val = newTemp   <e1>val = newTemp   <e1>valType = newTemp.valType   <e1>valType = newTemp.valType   <e1>valType = newTemp.valType   <e1>valType = newTemp.valType   <e1>valType = newTemp   <e2>valType   <e2>v</e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e2></e1></e1></e1></e1></e1></e1></e1>	
41 42	<exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp valType = 'int' <e1>val = newTemp <e2>val); if newTemp1 = (cA&gt;val &gt;= cE2&gt;val); if newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp valType = 'int' <e1>val = newTemp <e1>valType = newTemp valType = newTempvalType   <a> <e1> tinewTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 1 goto L3 L2: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>val = newTemp <e1>valType = newTemp.valType    <a> <e1> tinewTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>valType = newTemp.valType     A&gt; <e1> tinewTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp1 goto L1 else L2 L1: newTemp1 = (cA&gt; val = <e2> val); if newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp1 goto L1 else L2 L1: newTemp1 = (cA&gt; val = <e2> val); if newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp valType = 'int' <e1>val = newTemp <e1>valType = newTemp <e1>valType = newTemp <e1>valType = newTemp <a1> lR = <l> lR + <a2> lR + 'newTemp = <l> lava   + <a2> val' <a1> val = newTemp <a1> valType = newTemp valType   if <i> <a2> type not match: ERROR <a1> lR = <i> lR + <a2> lR + 'newTemp = <i +="" <a2="" lava=""  =""> val' <a1> val = newTemp <a1> valType = newTemp valType</a1></a1></i></a2></i></a1></a2></i></a1></a1></a2></l></a2></l></a1></e1></e1></e1></e1></e2></e2></e1></e1></e1></a></e1></e1></e1></a></e1></e1></e2></e1></e1></e1></e1></e1></e1></e1></e1>	
41 42	<exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp.2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>-val = newTemp <e1>-valType = newTemp.valType <e1>-valType = newTemp.valType <e1>-valType = newTemp.valType <e1>-valType = newTemp.valType  <a>-<e1> type not match: ERROR <e1>-lR = <a>- lR + <e1>- lR +</e1></a></e1></e1></a></e1></e1></e1></e1></e1>	
41 42	<exprsn> <exprsn> <addexprsn></addexprsn></exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn> <addexprsn> != <exprsn></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1>val = newTemp <e1>valType = newTemp <e1>valType = newTemp <e1>\text{NewTemp.valType}  <a> &lt; E1&gt; type not match: ERROR <e1>\text{R} = &lt; A&gt; \text{R} + &lt; E1&gt; \text{R} + \text{NewTemp1} = (cA&gt; val) &gt; (eE2&gt; val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1> val = newTemp <e1> \text{NewTemp.valType} = 'int' <e1> \text{NewTemp.valType} = 'int' <e1> \text{NewTemp.valType} = 'int' \text{NewTemp1 goto L1 else L2} L1: newTemp1 = (cA&gt; val = \text{C2&gt; val); if newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp.valType = 'int' <e1> \text{val} = newTemp <e1> \text{Nal} = \text{val} \text{IR} + \text{val} \text{Val} \text{goto L3} L3: \text{newTemp.valType} = 'int' <e1> \text{val} = newTemp <e1> \text{Nal} = \text{val} \text{IR} + \text{val} \text{Val} \text{goto L3} L2: newTemp2 = 1 goto L3 L2: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L3: \text{Nal} = \text{val} = \text{val} \text{val} \text{inewTemp1 goto L1 else L2} L1: newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: \text{Nal} = \text{val} = \text{val} \text{val} \text{val} = \text{val} = \text{val} \text{val} \text{val} = \text{val} = \text{val} \text{val} \text{val} = \text{val} = \</e1></e1></e1></e1></e1></e1></e1></e1></e1></a></e1></e1></e1></e1>	
41 42 44	<exprsn> <addexprsn> <addexprsn></addexprsn></addexprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn> <addexprsn> != <exprsn> <item> + <addexprsn> <item> - <addexprsn></addexprsn></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3  L3: newTemp2 = 0 goto L3  L3: hewTemp valType = 'int' <e1> valType = newTemp <e1> valType = newTemp.valType  <a> <e1> type not match: ERROR <e1> L1: newTemp1 = (<a> val) = <e2> val); if newTemp1 = (<a> val) = <e2> val); if newTemp1 goto L1 else L2</e2></a></e2></a></e1></e1></a></e1></e1>	
41 42 43	<expren> <expren> <addexpren> <addexpren> <addexpren></addexpren></addexpren></addexpren></expren></expren>	<addexprsn> == <exprsn> <addexprsn> != <exprsn> <addexprsn> != <exprsn> <item> + <addexprsn> <item></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2:newTemp2 = 0 goto L3 L3:newTempxalType = 'int' <e!>val = newTemp <e!>val = val = xel = xel = yal); if newTemp1 = (xal &gt; xal &gt; xel ≥ xal); if newTemp1 = (xal &gt; xal &gt; xel ≥ xal); if newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L3: newTemp = 0 goto L3 L3: newTemp valType = 'int' <e!>val = newTemp <e!>val = newTemp <e!>val = newTemp <e!>val = newTemp  = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp = 0 goto L3 L3: newTemp = newTemp <e!>val = newTemp <a!>val = val + xal = xa</a!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!>	
41 42 43	<expren> <expren> <addexpren> <addexpren> <addexpren></addexpren></addexpren></addexpren></expren></expren>	<addexprsn> == <exprsn> <addexprsn> != <exprsn> <addexprsn> != <exprsn> <item> + <addexprsn> <item></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2:newTemp2 = 0 goto L3 L3:newTempvalType = 'int' <e!>val = newTemp  (A&gt;<e!>Val = val = <e2>val); if newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp valType = 'int' <e!>val = newTemp valType  <a><e!>val = newTemp <e!>val = newTemp valType = newTemp.valType  L1: newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp valType = 'int' <e!>val = newTemp <a!>val = newTemp <a!>va</a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></a!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></a></e!></e2></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!></e!>	
41 42 45 46	<exprsn> <exprsn> <addexprsn> <addexprsn> <item></item></addexprsn></addexprsn></exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn> <addexprsn>  = <exprsn> <item> + <addexprsn> <item> + <addexprsn> <item> + <addexprsn></addexprsn></item></addexprsn></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp2 = 0 goto L3 L3: newTemp2 = i'nt' <e1>val = newTemp <e1>val : newTemp1 goto L1 else L2 L1: newTemp1 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp2 = 0 goto L3 L3: newTemp valType = 'int' <e1>val = newTemp <e1>valType = newTempvalType  <a><e1>type not match: ERROR <e1>tyle = newTempvalType  I1: newTemp1 goto L1 else L2 L1: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp4 goto L1 else L2 L1: newTemp1 goto L1 else L2 L1: newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L3: newTemp2 = 1 goto L3 L2: newTemp2 = 1 goto L3 L3: newTemp4 = 'A&gt;-\text{L1: newTemp1 goto L1 else L2} L1: newTemp1 goto L1 else L2 L1: newTemp2 = 1 goto L3 L3: newTemp4 = 'A&gt;-\text{L2: newTemp2 else L2} L1: newTemp2 = newTemp  <e1>val = newTemp  <e1>val = newTemp  <a>-L3: Npe not match: ERROR</a></e1></e1></e1></e1></a></e1></e1></e1></e1></e1></e1></e1></e1></e1></e1></e1>	
40 41 42 43 44 45	<exprsn> <addexprsn> <addexprsn> <addexprsn> <item></item></addexprsn></addexprsn></addexprsn></exprsn>	<addexprsn> == <exprsn> <addexprsn> == <exprsn> <addexprsn> != <exprsn> <item> + <addexprsn> <item> <factor> * <item> <factor> / <item></item></factor></item></factor></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	Qoto L3     L2: newTemp2 = 0     goto L3     L3: newTempy arType = int'     < E1 > valType = newTemp vaType	
40 41 42 43 44 45 46 49 50	<exprsn> <exprsn> <addexprsn> <addexprsn> <addexprsn> <item> <item> <item> <factor> <factor></factor></factor></item></item></item></addexprsn></addexprsn></addexprsn></exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn> <addexprsn> l= <exprsn> <item> + <addexprsn> <item> - <addexprsn> <item> inum  inum</item></addexprsn></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	Qoto L3     L2: newTempy = 0     Qoto L3     L3: newTempy = int'     <	
40 41 42 42 43 44 45 46 47	<exprsn> <exprsn> <addexprsn> <addexprsn> <item> <item> <factor> <factor> <factor></factor></factor></factor></item></item></addexprsn></addexprsn></exprsn></exprsn>	<addexprsn> == <exprsn> <addexprsn> == <exprsn> <addexprsn> l= <exprsn> <item> + <addexprsn> <item> - <addexprsn> <item> inum  inum  fnum</item></addexprsn></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp valType = int'	
40 41 42 43 44 45 46 49 50	<exprsn> <exprsn> <addexprsn> <addexprsn> <addexprsn> <item> <item> <item> <factor> <factor></factor></factor></item></item></item></addexprsn></addexprsn></addexprsn></exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn> <addexprsn> l= <exprsn> <item> + <addexprsn> <item> - <addexprsn> <item> inum  inum</item></addexprsn></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	200 L3   L2:newTemp2 = 0   goto L3   L3:newTemp.valType = int'	
40 41 42 43 44 45 46 47	<exprsn> <exprsn> <addexprsn> <addexprsn> <addexprsn> <item> <item> <item> <factor> <factor> <factor> <factor></factor></factor></factor></factor></item></item></item></addexprsn></addexprsn></addexprsn></exprsn></exprsn>	<addexprsn> &gt;= <exprsn> <addexprsn> == <exprsn> <addexprsn> != <exprsn> <item> - <addexprsn> <item> - <addexprsn> <item> - <addexprsn> <item> - <addexprsn>  inum  inum  fnum  fnum</addexprsn></item></addexprsn></item></addexprsn></item></addexprsn></item></exprsn></addexprsn></exprsn></addexprsn></exprsn></addexprsn>	goto L3 L2: newTemp2 = 0 goto L3 L3: newTemp valType = init'	
40 41 42 43 44 45 46 47 50 51	<exprsn> <addexprsn> <addexprsn> <addexprsn> <item> <item> <item> <factor> <factor> <factor> <factor> <factor> <factor> <pactor> <pactor< p=""> <pactor> <pactor< p=""> &lt;</pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor></pactor<></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></factor></factor></factor></factor></factor></factor></item></item></item></addexprsn></addexprsn></addexprsn></exprsn>	<pre><addexpren> &gt;= <expren>  <addexpren> != <expren>  <item> + <addexpren>  <item></item></addexpren></item></expren></addexpren></expren></addexpren></pre>	20	
40 41 42 43 44 45 46 47 50 51	<exprsn> <addexprsn> <addexprsn> <addexprsn> <item> <item> <item> <factor> <factor> <factor> <factor> <factor> <factor> <pactor> <pactor< p=""> <pactor> <pactor< p=""> &lt;</pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor<></pactor></pactor<></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></factor></factor></factor></factor></factor></factor></item></item></item></addexprsn></addexprsn></addexprsn></exprsn>	<pre><addexpren> &gt;= <expren>  <addexpren> != <expren>  <item> + <addexpren>  <item></item></addexpren></item></expren></addexpren></expren></addexpren></pre>	Septic	
40 41 42 43 44 45 46 47 50 51 52	<exprsn> <exprsn> <addexprsn> <addexprsn> <addexprsn> <item> <item> <item> <factor> <factor> <factor> <factor> <factor></factor></factor></factor></factor></factor></item></item></item></addexprsn></addexprsn></addexprsn></exprsn></exprsn>	<pre><addexprsn> == <exprsn> <addexprsn> == <exprsn>   -<addexprsn>   Factor&gt; - <a d<="" dispress="" href="https://dispress/d&lt;/td&gt;&lt;td&gt;goto 1.3  L2: new Year year 1/9 pc = 'int' &lt;1-10 xull'ype = new Year year 1/9 pc = 'int' &lt;1-10 xull'ype = new Year year 1/9 pc &lt;1-10 xull'ype = n&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;40&lt;br&gt;41&lt;br&gt;42&lt;br&gt;42&lt;br&gt;43&lt;br&gt;44&lt;br&gt;45&lt;br&gt;46&lt;br&gt;47&lt;br&gt;50&lt;br&gt;51&lt;br&gt;52&lt;/td&gt;&lt;td&gt;&lt;Expren&gt;  &lt;Expren&gt;  &lt;AddExpren&gt;  &lt;AddExpren&gt;  &lt;AddExpren&gt;  &lt;Item&gt;  &lt;Item&gt;  &lt;Item&gt;  &lt;Factor&gt;  &lt;Addenote the prediction of the p&lt;/td&gt;&lt;td&gt;&lt;pre&gt;&lt;AddExpren&gt; &gt;= &lt;Expren&gt; &lt;AddExpren&gt; == &lt;Expren&gt; &lt;AddExpren&gt; != &lt;Expren&gt; &lt;Item&gt; + &lt;AddExpren&gt; &lt;Item&gt; + &lt;AddExpren&gt; &lt;Item&gt;  &lt;/td&gt;&lt;td&gt;goto L3 L2: newTemp2 = 0 L3:newTemp2 = 0 L3:newTemp2 = 0 L3:newTemp2 = 0 L3:newTemp2 = newTemp valType&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;40&lt;br&gt;41&lt;br&gt;42&lt;br&gt;43&lt;br&gt;44&lt;br&gt;45&lt;br&gt;46&lt;br&gt;47&lt;br&gt;50&lt;br&gt;51&lt;br&gt;52&lt;/td&gt;&lt;td&gt;&lt;Exprsn&gt; &lt;Exprsn&gt; &lt;AddExprsn&gt; &lt;AddExprsn&gt; &lt;AddExprsn&gt; &lt;Item&gt; &lt;Item&gt; &lt;Item&gt; &lt;Factor&gt; &lt;Factor&gt; &lt;Factor&gt; &lt;Factor&gt; &lt;Factor&gt;&lt;/td&gt;&lt;td&gt;&lt;pre&gt;&lt;AddExprsn&gt; == &lt;Exprsn&gt; &lt;AddExprsn&gt; == &lt;Exprsn&gt; &lt;/ddExprsn&gt; &lt;/tem&gt; -&lt;AddExprsn&gt; &lt;/tem&gt; Factor&gt; - &lt;a href=" https:="" td=""><td>goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 10* L3: newTemp3 = now Temp0 = now Temp0</td><td></td></a></addexprsn></exprsn></addexprsn></exprsn></addexprsn></pre>	goto L3 L2: newTemp2 = 0 goto L3 L2: newTemp2 = 10* L3: newTemp3 = now Temp0	
40 41 42 42 43 44 45 46 47 50 51 52	<expren> <expren> <addexpren> <addexpren> <addexpren> <item> <item> <item> <factor> <addenote of="" p<="" prediction="" td="" the=""><td><pre><addexpren> &gt;= <expren> <addexpren> == <expren> <addexpren> != <expren> <item> + <addexpren> <item> + <addexpren> <item>  </item></addexpren></item></addexpren></item></expren></addexpren></expren></addexpren></expren></addexpren></pre></td><td>goto L3 L2: new Firmp2 = 0 Qoto L3 L3: new Firmpy = Int'  Ab &lt; E1&gt; val Type = new Firmp val Type  Ab &lt; E1&gt; kpe not match: ERROR</td><td></td></addenote></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></item></item></item></addexpren></addexpren></addexpren></expren></expren>	<pre><addexpren> &gt;= <expren> <addexpren> == <expren> <addexpren> != <expren> <item> + <addexpren> <item> + <addexpren> <item>  </item></addexpren></item></addexpren></item></expren></addexpren></expren></addexpren></expren></addexpren></pre>	goto L3 L2: new Firmp2 = 0 Qoto L3 L3: new Firmpy = Int'  Ab < E1> val Type = new Firmp val Type  Ab < E1> kpe not match: ERROR	
40 41 42 43 44 45 46 47 48 49 50 51 52	<expren> <expren> <addexpren> <addexpren> <addexpren> <item> <item> <item> <factor> <factor <factor=""> <factor> &lt;</factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></factor></item></item></item></addexpren></addexpren></addexpren></expren></expren>	<pre></pre>	12   12   12   13   13   13   14   15   15   15   15   15   15   15	
40 41 42 43 44 45 46 47 48 49 50 51 52 53	<expren> <expren> <addexpren> <addexpren> <addexpren> <item> <item> <item> <factor> <factor> <factor> <factor> <factor> <factor> <factor> <factor> <pactor> <pactor< p=""> <actor> <pactor< p=""> <actor> <pactor< p=""> <actor< p=""></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></actor<></pactor<></actor></pactor<></actor></pactor<></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></pactor></factor></factor></factor></factor></factor></factor></factor></factor></item></item></item></addexpren></addexpren></addexpren></expren></expren>	-AddExprano >= <exprano +="" -addexprano="" -itamo="" <addexprano="" <exprano="" <itamo="" <itamo<="" td=""  ="&lt;Exprano"><td>golo L3 L2: renviremp2 = 0 golo L3 L3: renviremp2 = 10 golo L3 L3: renviremp3 = 10 cd by a renyine - Intitive Callo val - Intitiv</td><td></td></exprano>	golo L3 L2: renviremp2 = 0 golo L3 L3: renviremp2 = 10 golo L3 L3: renviremp3 = 10 cd by a renyine - Intitive Callo val - Intitiv	
40 41 42 43 44 45 46 47 48 49 50 51 52 53	<expren> <addexpren> <addexpren> <addexpren> <item> <item> <item> <item> <factor> <factor> <factor> <factor> <actualargs> <actualargs></actualargs></actualargs></factor></factor></factor></factor></item></item></item></item></addexpren></addexpren></addexpren></expren>	<pre><addexpren> = <expren> <addexpren> = <expren> <addexpren> = <expren> <addexpren> = <expren> <item> + <addexpren> <item> + <addexpren> <item> + Factor&gt; / <item></item></item></addexpren></item></addexpren></item></expren></addexpren></expren></addexpren></expren></addexpren></expren></addexpren></pre>	goto 13  12: new Temp2 = 0  13: new John John John John John John John John	