

Non-Terminals	hold	comment	PrimaryMission	struct	unit	digit	joe	company	response	inquire	go	phase	miss
<program>	1	1	1	1	1	1	1	1	1	1			1
<Prod_comment>	3	2, 3	3	3	3	3	3	3	3	3	3	3	3
<GlobDec>	7, 8	8	8	6, 8	4, 8	4, 8	4, 8	4, 8	4, 8	8	8	8	5, 8
<Declare1>													
<functRet1>													
<struct1>													
<DeclareOption>	16			15	13	13	13	13	13				14
<Declare>													
<functRet>													
<struct>													
<const>													
<Literal>													
<Literal2>													
<datatype>					32	33	34	35	36				
<DeclareChoice>													
<InitChoice>													
<addID>													
<N1>													
<N2>													
<index>													
<Smath>													
<arrayAID>													
<ElemChoice>													
<Element>													

<addElem>													
<M_Elem>													
<M2_Elem>													
<memDec>					61	61	61	61	61				
<initDec>													
<initDecChoice>													
<dtypeA>					68	68	68	68	68				
<ExdtypeA>													
<returnParam>													
<main>			75										
<body>	76	85		76, 80	76	76	76	76	76	79	82	83	76
<print>													
<postval>													
<ConcatLit>													
<ExConcatLit>													
<scan>													
<scanVal>													
<addScan>													
<ExtaddScan>													
<assignChoice>				106									
<AccessAssignDtype>													
<assignValueChoice>													
<assignmentInit>													
<structInitial>													
<AccessValueChoice>													

<structCall>													
<varStruct>													
<StructArray>													
<addStructvar>													
<functCall>													
<param>													
<addparam>													
<initStruct>													
<AssignSym>													
<oper1>													
<assignValue>													
<mntCond>													
<mnt>													
<for>										145			
<vall>													
<valPP>													
<operator>													
<RelOp>													
<RelopExt>													
<opl>													
<LogOper>													
<ifelse>													
<ifcondition>													
<break>													
<LogOp>													

<ExtLogOp>													
<elseif>	174	174		174	174	174	174	174	174	174	174	174	174
<else>	176	176		176	176	176	176	176	176	176	176	176	176
<do_while>											177		
<while>											178		
<switch>													
<case>													
<default>													
<MathOp>	185	185		185	185	185	185	185	185	185	185	185	185
<operCond>													
<operand>													
<OperationMath>													
<structMath>													
<operExt_s>													
<simMathOp>													
<S_MathExt>													
<operCondExt>													
<end>													

Non-Terminals	post	capture	inorder	order	otherorder	action	operation	campaign	deploy	abort	backup
<program>											
<Prod_comment>	3	3	3					3	3	3	3
<GlobDec>	8	8	8					8			
<Declare1>											
<functRet1>											
<struct1>											
<DeclareOption>											
<Declare>											
<functRet>											

<struct>											
<const>											
<Literal>											
<Literal2>											
<datatype>											
<DeclareChoice>											
<InitChoice>											
<addID>											
<N1>											
<N2>											
<index>											
<Smath>											
<arrayAID>											
<ElemChoice>											
<Element>											
<addElem>											
<M_Elem>											
<M2_Elem>											
<memDec>											
<initDec>											
<initDecChoice>											
<dtypeA>											
<ExdtypeA>											
<returnParam>											
<main>											

<body>	77	78	81					84		86	86
<print>	87										
<postval>											
<ConcatLit>											
<ExConcatLit>											
<scan>		97									
<scanVal>											
<addScan>											
<ExtaddScan>											
<assignChoice>											
<AccessAssignDtype>											
<assignValueChoice>											
<assignmentInit>											
<structInitial>											
<AccessValueChoice>											
<structCall>											
<varStruct>											
<StructArray>											
<addStructvar>											
<functCall>											
<param>											
<addparam>											
<initStruct>											
<AssignSym>											
<oper1>											

<assignValue>											
<mntCond>											
<mnt>											
<for>											
<vall>											
<valPP>											
<operator>											
<RelOp>											
<RelopExt>											
<opl>											
<LogOper>											
<ifelse>			166								
<ifcondition>											
<break>										169	
<LogOp>											
<ExtLogOp>											
<elseif>	174	174		174	173			174		174	174
<else>	176	176	176	175				176		176	176
<do_while>											
<while>											
<switch>								179			
<case>						181	180				
<default>						182					
<MathOp>	185	185	185					185		185	185
<operCond>											

<operand>											
<OperationMath>											
<structMath>											
<operExt_s>											
<simMathOp>											
<S_MathExt>											
<operCondExt>											
<end>									203		

Non-Terminals	id	Numli t	Decli t	Charli t	Stringli t	AFFIRMATIV E	NEGATIV E	+	-	/	%	^	*	++	--
<program>															
<Prod_comment>	3													3	3
<GlobDec>	8													8	8
<Declare1>															
<functRet1>															
<struct1>															
<DeclareOption>															
<Declare>															
<functRet>															
<struct>															
<const>															
<Literal>		22	23	24	25	26	27								
<Literal2>		28	29	30	31										
<datatype>															
<DeclareChoice>															
<InitChoice>															
<addID>															
<N1>															
<N2>															

<index>	48	47													
<Smath>								4 9	4 9	4 9	4 9	4 9	4 9		
<arrayAID>															
<ElemChoice>		53	53	53	53										
<Element>		55	55	55	55										
<addElem>															
<M_Elem>															
<M2_Elem>															
<memDec>															
<initDec>															
<initDecChoice>															
<dtypeA>															
<ExdtypeA>															
<returnParam>	73	72	72	72	72	72	72								
<main>															
<body>	80													80	80
<print>															
<postval>	89	88	88	88	88	88	88								
<ConcatLit>															
<ExConcatLit>															
<scan>															
<scanVal>															
<addScan>															
<ExtaddScan>															

<assignChoice>	104, 105													105	105
<AccessAssignDtype>	107														
<assignValueChoice>															
<assignmentInit>															
<structInitial>								115	115	115	115	115	115		
<AccessValueChoice>								116	116	116	116	116	116		
<structCall>															
<varStruct>															
<StructArray>															
<addStructvar>															
<functCall>															
<param>	125														
<addparam>															
<initStruct>	130	129	129	129	129										
<AssignSym>								131	131	131	131	131	131		
<oper1>								132	133	134	135	136	137		
<assignValue>	140	139	139	139	139	139	139								
<mntCond>	142													141	141
<mnt>														143	144
<for>															
<vall>	148	146	147					149	149	149	149	149	149		

<valPP>															
<operator>								151	152	153	154	155	156		
<RelOp>	157														
<RelopExt>															
<opl>															
<LogOper>															
<ifelse>															
<ifcondition>	167														
<break>															
<LogOp>															
<ExtLogOp>															
<elseif>	174													17 4	17 4
<else>	176													17 6	17 6
<do_while>															
<while>															
<switch>															
<case>	181														
<default>															
<MathOp>	184	184	184												
<operCond>	187	187	187												
<operand>	190	188	189												
<OperationMath>															
<structMath>								194	194	194	194	194	194		
<operExt_s>								195	195	195	195	195	195		
<simMathOp>	197	197	197												

<S_MathExt>								198	198	198	198	198	198		
<operCondExt>								201	201	201	201	201	201		
<end>															

Non-Terminals	=	>=	<=	==	!=	()	[]	{	}	:	,	.	;		&	#
<program>																		
<Prod_comment>							3				3							
<GlobDec>							8											
<Declare1>	9					10							9	9	9			
<functRet1>						11				12								
<struct1>																		
<DeclareOption>																		
<Declare>	17					18							17	17	17			
<functRet>						19												
<struct>										20								
<const>	21																	
<Literal>																		
<Literal2>																		
<datatype>																		
<DeclareChoice>	37							38					37					
<InitChoice>	40												39		41			
<addID>													42		43			
<N1>								44										
<N2>	46						46	45					46	46	46			
<index>																		
<Smath>									49									
<arrayAID>	51														52			
<ElemChoice>										54								
<Element>																		

<addElem>											57		56				
<M_Elem>													58				
<M2_Elem>											60		59				
<memDec>											62						
<initDec>								64					63		65		
<initDecChoice>													66		67		
<dtypeA>							69										
<ExdtypeA>							71						70				
<returnParam>							74										
<main>																	
<body>											86						
<print>																	
<postval>																	
<ConcatLit>					91	94	93						90	92			
<ExConcatLit>						96							95				
<scan>																	
<scanVal>																	98
<addScan>							101						99	100			
<ExtaddScan>							103						102				
<assignChoice>																	
<AccessAssignDtype>																	
<assignValueChoice>	109					110		108						111	110		
<assignmentInit>	112													113			
<structInitial>	114, 115													115			
<AccessValueChoice>	116													116			

<structCall>																	
<varStruct>																	
<StructArray>							120	119					120		120		
<addStructvar>													121		122		
<functCall>						123, 124	124								124		
<param>							126										
<addparam>							128						127				
<initStruct>																	
<AssignSym>																	
<oper1>	138																
<assignValue>																	
<mntCond>																	
<mnt>																	
<for>																	
<vall>																	
<valPP>															150		
<operator>																	
<RelOp>																	
<RelopExt>		158	158	158	158		159								159		
<opl>		161	162	160	163												
<LogOper>																164	165
<ifelse>																	
<ifcondition>						168											
<break>																	
<LogOp>						170	172									171	171

<ExtLogOp>																		
<elseif>																		
<else>																		
<do_while>																		
<while>																		
<switch>																		
<case>																		
<default>											183							
<MathOp>						184					185							
<operCond>						186												
<operand>																		
<OperationMath>						192								191				
<structMath>	194					194	194							193	194			
<operExt_s>																		
<simMathOp>																		
<S_MathExt>						199	200								200			
<operCondExt>															202			
<end>																		