|  |  |  |  |
| --- | --- | --- | --- |
| 1 | <program> | → | <const> <globalDec> <main> <Localdec> <body> <end> |
| 2 | <const> | → | hold <consDec>;<const> |
| 3 | <const> | → | Λ |
| 4 | <consDec> | → | <constUnit> <constDigit> <constJoe> <constCompany> <constResponse> |
| 5 | <constUnit> | → | unit idc1 = number |
| 6 | <constUnit> | → | Λ |
| 7 | <constDigit> | → | digit idc2 = decimal |
| 8 | <constDigit> | → | Λ |
| 9 | <constJoe> | → | joe idc3 = “ SingChar ” |
| 10 | <constJoe> | → | Λ |
| 11 | <constCompany> | → | company idc4 = “ Characters ” |
| 12 | <constCompany> | → | Λ |
| 13 | <constResponse> | → | response idc5 = <bool> |
| 14 | <bool> | → | AFFIRMATIVE |
| 15 | <bool> | → | NEGATIVE |
| 16 | <constResponse> | → | Λ |
| 17 | <globalDec> | → | <unitType> <joeType> <digitType> <companyType> <responseType> |
| 18 | <globalDec> | → |  |
| 19 | <unitType> | → | unit id2 <unitInit>; <unitType> |
| 20 | <unitType> | → | Λ |
| 21 | <unitInit> | → | = number <addInit> |
| 22 | <unitInit> | → | Λ |
| 23 | <addInit> | → | , <unitInit> |
| 24 | <addInit> | → | Λ |
| 25 | <unitInit> | → | Λ |
| 26 | <joeType> | → | joe id3 <joeInit>; <joeType> |
| 27 | <joeType> | → | Λ |
| 28 | <joeInit> | → | = “ SingChar ” <addjoe> |
| 29 | <addjoe> | → | , <joetype> |
| 30 | <joeInit> | → | Λ |
| 31 | <addjoe> | → | Λ |
| 32 | <digitType> | → | digit id4 <digitInit>; <digitType> |
| 33 | <digitType> | → | Λ |
| 34 | <digitInit> | → | = decimal <addDigit> |
| 35 | <digitInit> | → | Λ |
| 36 | <addDigit> | → | Λ |
| 37 | <addDigit> | → | , <digitType> |
| 38 | <companyType> | → | company id5 <companyInit>; <companyType> |
| 39 | <companyType> | → | Λ |
| 40 | <companyInit> | → | = “ Characters “ <addCompany> |
| 41 | <addCompany> | → | , <companyType> |
| 42 | <companyInit> | → | Λ |
| 43 | <addCompany> | → | Λ |
| 44 | <responseType> | → | response id6 <responseInit>; <responseType> |
| 45 | <responseType> | → | Λ |
| 46 | <responseInit> | → | = bool <addResponse> |
| 47 | <addResponse> | → | , <responseType> |
| 48 | <responseInit> | → | Λ |
| 49 | <addResponse> | → | Λ |
| 50 | <main> | → | PrimaryMission(){ |
| 51 | <Localdec> | → | <unitType> <joeType> <digitType> <companyType> <responseType> <arrType> <funct> <struct> |
| 52 | <arrType> | → | <arrUnit> <arrDigit> <arrCompany> <arrJoe> |
| 53 | <arrType> | → | Λ |
| 54 | <arrUnit> | → | unit <N1> <N2> <unitAID>; <arrUnit> |
| 55 | <arrUnit> | → | Λ |
| 56 | <N1> | → | [<index1>] |
| 57 | <N2> | → | [<index2>] |
| 58 | <index1> | → | numbers |
| 59 | <index2> | → | numbers |
| 60 | <N1> | → | Λ |
| 61 | <N2> | → | Λ |
| 62 | <unitAID> | → | = {numbers} |
| 63 | <unitAID> | → | Λ |
| 64 | <arrDigit> | → | digit <N1> <N2> <digitAID>; <arrDigit> |
| 65 | <arrDigit> | → | Λ |
| 66 | <digitAID> | → | = {decimal} |
| 67 | <digitAID> | → | Λ |
| 68 | <arrCompany> | → | company <N1> <N2> <companyAID>; <arrCompany> |
| 69 | <arrCompany> | → | Λ |
| 70 | <companyAID> | → | = { “letters” } |
| 71 | <companyAID> | → | Λ |
| 72 | <arrJoe> | → | joe <N1> <N2> <joeAID>; <arrJoe> |
| 73 | <arrJoe> | → | Λ |
| 74 | <joeAID> | → | = { “char” } |
| 75 | <joeAID> | → | Λ |
| 76 | <funct> | → | <dtype> functName (<dtypeA>) {<statement>} |
| 77 | <dtype> | → | Λ |
| 78 | <dtype> | → | unit |
| 79 | <dtype> | → | company |
| 80 | <dtypeA> | → | <dtype> idx1 |
| 81 | <dtypeA> | → | <dtype>idx1, <dtype> idx2 |
| 82 | <dtypeA> | → | Λ |
| 83 | <struct> | → | struct structName { <SDec> } <structvar>; |
| 84 | <struct> | → | Λ |
| 85 | <sDec> | → | <sDtype> idxS <index>; <sDec> |
| 86 | <sDec> | → | Λ |
| 87 | <index> | → | [numbers] |
| 88 | <Index> | → | Λ |
| 89 | <structvar> | → | structN , <structvar> |
| 90 | <structvar> | → | Λ |
| 91 | <body> | → | <print> <scan> <for> <ifelse> <do-while> <while> <switch> |
| 92 | <print> | → | post(<postval>); <print> |
| 93 | <print> | → | Λ |
| 94 | <postval> | → | “Upperchar/Lowerchar” |
| 95 | <postval> | → | Idx1 |
| 96 | <scan> | → | captured(%<cType>, &idx2); <scan> |
| 97 | <scan> | → | Λ |
| 98 | <cType> | → | d |
| 99 | <cType> | → | f |
| 100 | <cType> | → | s |
| 101 | <for> | → | inquire(idx = <val1>; idx RelOp <val1>; idx<mnt>){ <statement>} |
| 102 | <for> | → | Λ |
| 103 | <val1> | → | num |
| 104 | <val1> | → | zero |
| 105 | <mnt> | → | ++ |
| 106 | <mnt> | → | -- |
| 107 | <ifelse> | → | Inorder(<condition>){ <statement> } <elseif> <else> <ifelse> |
| 108 | <ifelse> |  | Λ |
| 109 | <elseif> | → | otherorder(<condition>){ <statement> } <elseif> |
| 110 | <elseif> | → | Λ |
| 111 | <else> | → | other{ <statement> } |
| 112 | <else> | → | Λ |
| 113 | <do-while> | → | go{ <statement> }phase(<condition>); <do-while> |
| 114 | <do-while> | → | Λ |
| 115 | <while> | → | phase(<condition>){ <statement> } <while> |
| 116 | <while> | → | Λ |
| 117 | <switch> | → | campaign(){ <case> } |
| 118 | <case> | → | operation idn: <statement> abort; <case> |
| 119 | <switch> | → | Λ |
| 120 | <case> | → | Λ |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |