START -> OUTER MAIN OUTER

**********Outer Part of Main**********

OUTER -> e |

COMMENT OUTER |

VARDEF OUTER |

STRUCT OUTER |

FUNCTION OUTER |

MACRO OUTER

**********MACRO n COMMENT**********

MACRO -> #define IDENTIFIER TEXT //TEXT doesn't allow whitespace

COMMENT -> /* TEXT1 */ | //TEXT2 //TEXT1 doesn't allow */ and TEXT2 doesn't allow \n

VOID -> void

INT -> int

MAINARG -> e |

int argc, char* argv[]

MAIN -> VOID main(MAINARG){INNER} |

int main(MAINARG){INNER}

**********TYPES**********

TYPE -> int | bool | void | float | double | struct IDENTIFIER | TYPE* | char

STRUCT -> struct INDENTIFIER{MVARDEF VARDEF;};

**********Multiple Variable Definition**********

MVARDEF -> MVARDEF VARDEF;|e

FUNCTION -> TYPE VARNAME(FARG){INNER}

**********Inner Part of Function**********

INNER -> COMMENT INNER |

LOOP INNER |

CONDITIONAL INNER |

VARDEF INNER |

STRUCT INNER |

FCALL INNER |

RETURN INNER |

;|

e |

INPUT INNER |

OUTPUT INNER |

MATH INNER |

ASSIGN INNER

**********Variable Names**********

VARNAME -> *VARNAME | &VARNAME | IDENTIFIER BRACKET

BRACKET -> BRACKET[INTMATH] | e

MARG -> MARG TYPE VARNAME, | e

FARG -> MARG TYPE VARNAME

**********Variable Definitions**********

VARDEF ->TYPE MVAR VARNAME; |

TYPE MVAR VARNAME = CONST; |

TYPE MVAR VARNAME = FCALL; |

TYPE MVAR VARNAME = RMATH; |

TYPE MVAR VARNAME = VARNAME;|

TYPE MVAR VARNAME = {MCONST CONST};

MVAR -> MVAR VARNAME, |

MVAR VARNAME = CONST, |

MVAR VARNAME = FCALL, |

MVAR VARNAME = RMATH, |

MVAR VARNAME = VARNAME, |

MVAR VARNAME = MCONST, |

e

MCONST -> MCONST CONST,|e

OPERATOR -> +|

-|

*|

/|

&|

^|

||

%

**********Maths**********

MATH -> VARNAME = RMATH; |

;|

TYPE VARNAME = RMATH; |

VARNAME OPERATOR= RMATH; |

VARNAME++; |

VARNAME--; |

++VARNAME; |

--VARNAME;

**********RHS of Math**********

RMATH = VARNAME|

FCALL|

VARNAME++ |

VARNAME-- |

++VARNAME |

--VARNAME|

(RMATH)|

!RMATH |

CONST|

RMATH OPERATOR RMATH

**********INTMATH for Array Indexing**********

INTMATH -> VARNAME|

FCALL|

(INTMATH) |

!INTMATH |

INTCONST|

INTMATH OPERATOR INTMATH|

VARNAME++ |

VARNAME-- |

++VARNAME |

--VARNAME|

**********Conditional**********

CONDITIONAL -> IFN | SWITCHN

IFN -> if(COND){INNER}ELSE

RELATIONALOPERATOR ->>|

<|

==|

!=|

<=|

>=

COND -> (COND) |

COND&&COND |

COND||COND |

!COND |

RMATH |

COND RELATIONALOPERATOR COND

ELSE -> elif (COND) {INNER} ELSE | else {INNER} | e

SWITCHN -> switch(COND){SWITCHINNER}

SWITCHINNER -> case CONST:{INNER} SWITCHINNER |

case CONST: INNER SWITCHINNER |

default:{INNER} WODEFAULT|

default: INNER WODEFAULT|

e

WODEFAULT -> case CONST:{INNER} WODEFAULT |

case CONST: INNER WODEFAULT|

e

ARGT -> MARGT RMATH

MARGT -> MARGT RMATH,| e

FCALL -> VARNAME(ARGT)

**********Iteration**********

LOOP -> FOR | WHILE

FL1-> MATH | e

FL2-> COND | e

FL3 -> VARNAME = RMATH |

TYPE VARNAME = RMATH|

VARNAME OPERATOR= RMATH |

VARNAME++|

VARNAME--|

++VARNAME|

--VARNAME

FOR -> for(FL1 FL2; FL3){INNER}

WHILE->while(COND){INNER}

DOWHILE -> do{INNER}while(COND);

**********Input-Output**********

MIN -> MIN >>VARNAME |e

MOUT -> MOUT <<VARNAME |

MOUT << CONST |

MOUT << WHITESPACE |

e

INPUT -> in MIN >> VARNAME;

OUTPUT -> out MOUT << VARNAME;|

out MOUT << CONST;|

out MOUT << WHITESPACE;