Simple Program of Military Compiler

#acquire <headern>; //<decl> -> #acquire headn ; <decl>

// <decl> -> λ

#standby <idn> ; // <dec> -> #standby idn <standn>; <dec>

// <dec> -> λ

// <standn> -> λ

PrimaryMission() // <main> -> PrimaryMission() { <statements> }

{

//clrscr();

//<statements> -> <clears> commence();

// <body> <end>

//<body> -> <dtype> id <init> ; <dtype>

// <dtype> ->unit

// <dtype> -> company

// <dtype> -> digit

// <dtype -> joe

// <dtype> -> λ

//<init> = <value>

// <init> -> λ

}

**LOOP CFG**

<for> → inquire (<dtype> <idn> = <value> ; <idn> <logOp> <value> ; <idn> <unOp> ) { <statement> <for> }

<for> → λ

<logOp> → <=

<logOp> → >=

<unOp> → ++

<unOp> → --

<while> → phase (<condition>) {

<statement> <while>

}

<while> → λ

<do> → go {

<statement>

<do>

} phase (<condition>);

<do> → λ

<if> → inorder (<condition>) {

<statement> <cont>

} <cond> <if>

<if> → λ

<cont> → roll();

<cont> → λ

<cond> → otherorder (<condition>){ <statement> }

<cond> → order { <statement> }

<cond> → λ

<switch> → campaign(<condition>) { <case> }

<case> → operation <idn> : <statement> <def>

<def> → auto : <statement>

<def> → λ

**Input and Output**

<print> → post (<pstatement>); <print>

<print> → λ

<pstatement> → “ <strstatement> “

<pstatement> → <idn>

<pstatement> → λ

<scan> → captured (sstatement); <scan>

<sstatement> → “ %<ph> ”, <idn>

<ph> → d

<ph> → f

<ph> → c