**FORTE LANGUAGE**

**CONTEXT FREE GRAMMAR**

**MAIN:**

<main> 🡪 <pre\_main> global DT head() { <body>} <post\_main> $

<pre\_main>🡪 <function> <pre\_main>| <func\_sign> <pre\_main>| <struct><pre\_main> |<G\_Decl> <pre\_main> |epsilon

<post\_main>🡪 <function ><postmain>|epsilon

**BODY:**

<body>🡪 <mst>

< mst >🡪 <sst> < mst> |epsilon

<sst>🡪 <Decl> | < if\_else> | <whilst> |<do\_whilst>| <func\_call> | <return> |<assignment>

**RETURN:**

<return>🡪return int\_const;

**ARRAY & DECLARATION:**

<Decl>🡪 DT ID<Decl>’

<Decl>’🡪 <init><list> | [<size>]<A\_init><list>

<size>🡪 <ID\_int ><size>’ |epsilon

<size>’🡪 ,<ID\_int > | epsilon

<ID\_int>🡪 ID | int\_const

<init>🡪 assign\_op <init’> |epsilon

<init’>🡪 ID <init>|<const> | <E>

<A\_init>🡪 assign\_op {<A\_init’> |epsilon

<A\_init’>🡪<value>}|<A\_init2>}

<value>🡪<E><value1>|epsilon

<value1>🡪 ,<E><value1>|epsilon

<A\_init2> {<value>}<value2>|epsilon

<value2>🡪,{value>}<value2>|epsilon

<list>🡪 ;| ,ID<list’>

<list’>🡪<init><list>|[<size>]<A\_init><list>

<const>🡪 int\_const | float\_const | char\_const | string\_const

**CONDITION STATEMENT USED IN FOR/IF\_ELSE/WHILE:**

<OE>🡪<AND>< OE'>

<OE'>🡪 or\_op < AND >< OE'> | epsilon

<AND>🡪 <RE> < AND'>

<AND'> 🡪 and\_op <RE>< AND'> | epsilon

<RE>🡪<E><RE'>

<RE'> 🡪 Relat\_op <E><RE'> | epsilon

<E>🡪 <T><E'>

<E'>🡪 addsub\_op <T><E'> | epsilon

<T>🡪 <F><T'>

<T'>🡪 <MDM> <F><T'> | epsilon

<F>🡪 ID<F>’ | <const> | (<OE>) | incdec\_op ID | not\_op <F>

<F>’🡪 (<prmtrs2>) | incdec\_op |epsilon

<MDM>🡪 muldiv\_op | mode\_op

**PRE AND POST INCREMENT WHILE LOOP:**

<whilst>🡪whilst(<OE><AI>)<body2>

<AI>🡪;<inc>|;<assign>|epsilon

<assign>🡪 ID incdec\_assign\_op <ID\_int>

<inc>🡪 ID incdec\_op | incdec\_op ID

<body2> 🡪 ; | <sst> | { <mst> }

**IF\_ELSE:**

<if\_else>🡪 if(<OE>)<body2><op\_else>

<op\_else>🡪 otherwise<body2> | epsilon

**FUNCTIONS:**

<function> 🡪<type> ID(<prmtrs >) { <body>}

<type> 🡪 devoid | DT

< prmtrs >🡪 DT ID <m\_prmtrs> | epsilon

<m\_prmtrs> 🡪 , DT ID <m\_prmtrs> | epsilon

**FUNCTIONS SIGNATURE:**

<func\_sign> 🡪 <type> ID (<prmtrs >);

**FUNCTIONS CALLING:**

<func\_call> 🡪 ID (<prmtrs2>);

<prmtrs2>🡪<E><m\_prmtrs2> | epsilon

<m\_prmtrs2> 🡪 , <E> <m\_prmtrs2> | epsilon

**STRUCTURES:**

<struct> 🡪 struct ID { <Decl>} ;

**GLOBAL DECLARATION:**

<G\_delc> 🡪 <Decl>

**DO WHILE:**

<do\_whilst> 🡪 do{<body>} whilst(<OE>);

**ASSIGNMENT:**

<assignment>🡪 ID assign\_op < E> ;