S → vfunction S | types id sep S | ε

sep → function | dec2;   id → identifier

function → (dfun) { STM return E; }

types → int | long | short | float | double | char

dfun → types id2 N   N → ε | , dfun

id2 → * id | id   vfunction → void id (dfun) { STM }

normdec → op E   op → := | += | -= | /= | *=

declaration → types identifier dec | id normdec

dec2 → normdec | ε

STM → loop STM | declaration; STM | condition STM | ε

loop → while (comp) { STM } | for (dfor) { STM }

dfor → id=tmp; comp ; efor   efor → eefor id | id eefor

comp → E op2 E   eefor → ++ | --

E → T E'   E' → +T E' | -T E' | % T E' | ε

T → F T'   T' → * F T' | / F T' | ε

F → (E) E | id | num

condition → if (comp) { STM } cond

cond → else cond2 | ε

cond2 → condition | { STM }

op2 → == | >= | <= | < | > | !=

12. tmp → id | num

#	Nonterminal	first	follow
1	function	void types	, $, void, types, id
2			
3	dfun	types	)
4			
5	N	ε|,	)
6	id2	*|id	)|,
7			
8	types	int|long...|	*|id|(
9			
10	declaration	types|id	;
11	dec2	ε|op	;
12	normdec	op	;
13	op	=|+=|-=...	(|id|num
14	op2	==|>=|<=|>|<...	(|id|num
15			
16	loop	while|for	FIRST(STM)|}|return
17	comp	(|id|num	)|;
18			
19	dfor	id	)
20	efor	++|--|id	)
21	eefor	++|--	id|)
22	tmp	float|const	
23	num	float|const?|id	
24	condition	if	FIRST(STM)|}|return
25	cond	else|ε	}|return
A	z cond2	if|ε	}|return

1	STM	while|for|if	return |}	|types|id |ε
2	S	void|types|id		$
3	outin	printf|scanf		;
4				
5				
6	E	(|id|num	*|/|+|-|%|) |;
7	E′	+|-|%|ε	*|/|+|-|%|)
8	T	(|id|num	+|-|%|)|/|*
9	T′	*|/|ε	+|-|%|)|/|*
10	F	(|id|num	*|/|+|-|%|)|

12		void f types	types	id
13	S	function	declaration;
14		while for	if	types id
15	STM	loop	condition	declaration;
16		} return
17		ε	ε

20	E	(	num|id
21		TE′
22	E′	+	−	%	*|	/	)
23		+TE′	−TE′	%TE′	ε	ε	ε

	(	id	num	{T′	*	/
25	T
PARVAZ	FT′	FT′	FT′· }	*FT′	/FT′

				+	−	%	>
F	(id num				ε	ε	ε	ε
	(E)E id num							

op   void int | long! ---

Function   ftypes id (dfun) { STm freturn}

dfun   types id2 ddfun   id2 = ID
   int|long   N = ddfun
      id   *

ddfun   ,dfun   ε   { id2   id   *id

     int|long ---   return   }
types   self   freturn   return E;   ε
      types   id

declaration   types id dec2   id normdec
     op   ;
dec2   normdec   ε

     op
normdec   op E   { op   self { op2 self
    while   for
loop   while(comp){STm) for(dfor){STM}
     (id   num
Comp   Eop2 E
     id   id   ++ --
dfor   id = tmp; comp; efor   efor id eefor eeforid
     id num
        eefor self