

C1) This is a recursive descent parser. Write the grammar from this parser.

```

block()
    match('{')
    stmt()
    match('}')

stmt()
    if( currenttoken == 'id')
        stmt1()
        stmt()

stmt1()
    match('id')
    match('=')
    expr()
    match(';')

expr()
    match('id')
    exprs()

exprs()
    if( currenttoken == '+')
        match('+')
        exprs()

```

$block \rightarrow \{ stmt \}$
 $stmt \rightarrow stmt_1 stmt \mid \lambda$
 $stmt_1 \rightarrow id = expr ;$
 $expr \rightarrow id \ exprs$
 $exprs \rightarrow + \ exprs \mid \lambda$

C2) Given this grammar, compute First and Follow set, draw the parsing table

```

dcl = ID dcl2
dcl2 = ( formal ) stmt | [ NUM ]
formal = ID formals | empty
formals = , formal | empty

```

	first	follow
dcl	ID	\$
dcl2	(, [\$
formal	ID, λ)
formals	, , λ)

	\$	ID	()	stmt	[NUM]	,
dcl		dcl \rightarrow ID dcl2							
dcl2			dcl2 \rightarrow (formal) stmt			dcl2 \rightarrow [NUM]			
formal		formal \rightarrow ID formals		formal $\rightarrow \lambda$					
formals				formals $\rightarrow \lambda$					formals \rightarrow , formal