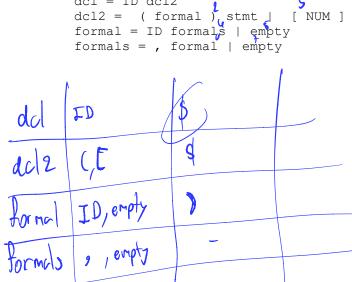
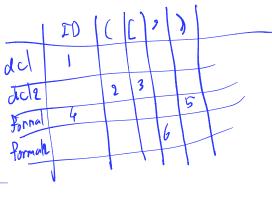
C1) This is a recursive descent parser. Write the grammar from this parser.

```
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()
```

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



dcl = ID dcl2



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()
```

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
```

