

```
parse-id-list()
```

```
{
```

```
}
```

```
parse-var-decl()
```

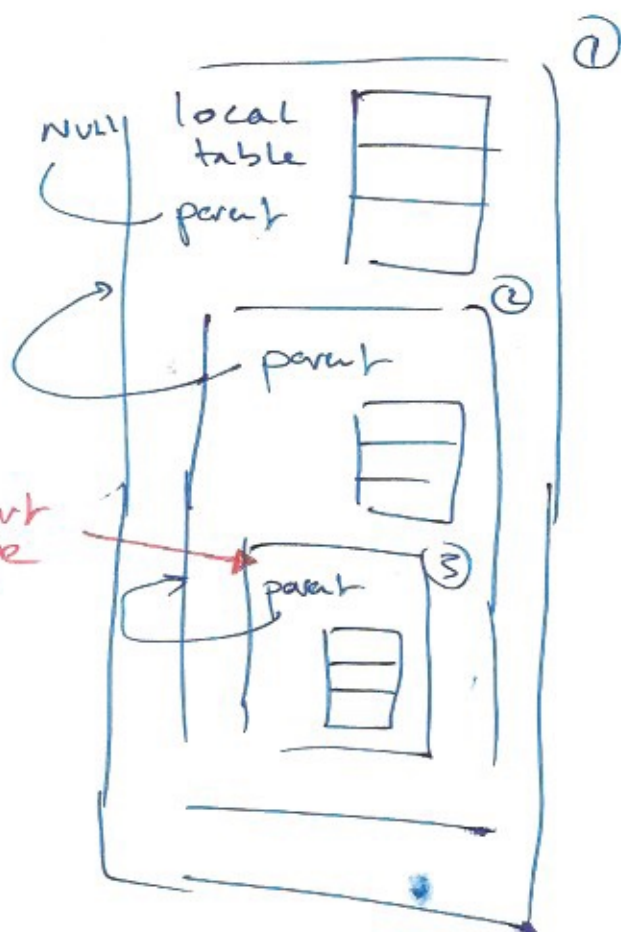
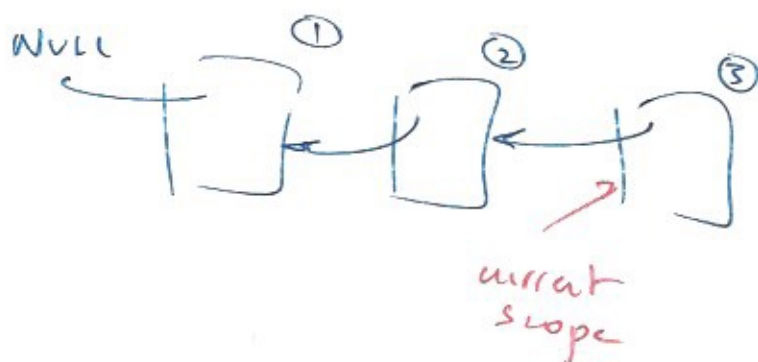
```
{ expect (VAR)
```

```
v-list = parse-id-list()
```

```
expect (COLON)
```

```
t-name = parse-type-name()  
}
```

```
VAR i, j : INT;
```



parse_id_list() returns pointer to id-list-node

{

tok = lexer.gettoken();

if (tok.token_type == ID)

else list = make_id_list_node(tok);

// error

tok = lexer.gettoken();

if (tok.token_type == COMMA)

{ list->next = parse_id_list();

return list;

else if (tok.token_type == COLON)

{ lexer.ungettoken(tok);

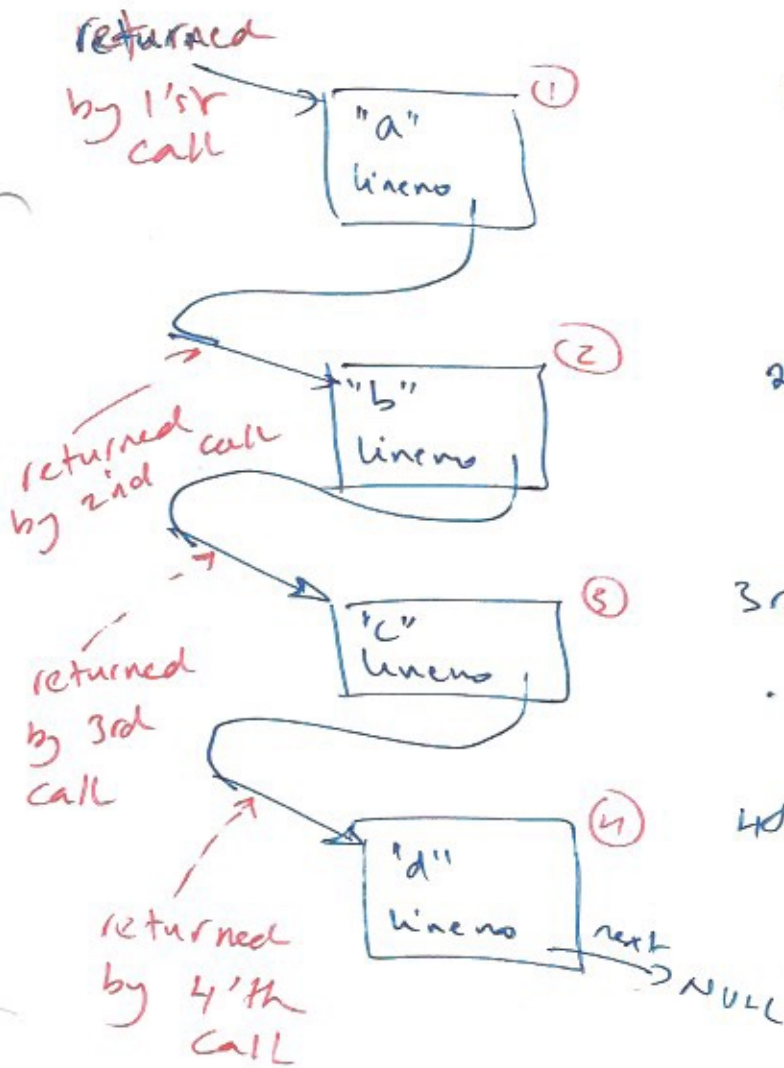
return list; // base case

} else

// error

}

a, b, c, d



1st call to parse_id_list()

2nd call to parse_id_list()

3rd call to parse_id_list()

4th call to parse_id_list()