Project1 Output Format

- **Syntactically Legal Program**

  Use indentation (2 spaces) to represent tree level

Project1 Output Format

- **Syntactically Legal Program**

For nonterminals, print “NAME (Lineno)”

```
Specifier (2)
```

(Lineno is first line of the nonterminal)

Project1 Output Format

- **Syntactically Legal Program**

For INT/FLOAT/CHAR terminals, print “NAME: attribute”

```
INT: 1
```

(For hex-int, output its decimal form, hex-char output its hex-form)

Project1 Output Format

- **Syntactically Legal Program**

  For ID terminals, print “ID: identifier”

```
ID: my_struct
```

Project1 Output Format

- **Syntactically Legal Program**

For TYPE terminals, print “TYPE: type”

```
TYPE: int
```

Project1 Output Format

- **Syntactically Legal Program**

For other terminals, print “NAME”

```
SEMI
```

Project1 Output Format

- **Lexical Error**

  Print the error type, with meaningful message

```
Error type A at Line #: (msg)
```

(only the type and lineno will be checked)

Project1 Output Format

- **Syntax Error**

Print the error type, with meaningful message

```
Error type B at Line #: (msg)
```

(only the type will be checked)

Project1 Output Format

- **Bonus - preprocessors**

  If you implement preprocessor, try to make it more powerful.

  For file inclusion, insert the file content as part of syntax tree.

  For macro, replace its usages (also the arguments).