

```
yacc -d lab4.y
lex lab4.l
gcc lex.yy.c y.tab.c symtable.c -o lab4
mgroover@knuth:~/compilers/lab4> ./lab4
int c;
Whitespace found
Letter found
return a token ;
cariage return

int y;
Whitespace found
Letter found
return a token ;
cariage return

int z;
Whitespace found
Letter found
return a token ;
cariage return

too many variable declarations
mgroover@knuth:~/compilers/lab4> █
```

If too many variables are attempted to be insert

```
mgroover@knuth:~/compilers/lab4> ./lab4
int x;
Whitespace found
Letter found
return a token ;
cariage return

int x;
Whitespace found
Letter found
return a token ;
cariage return

variable already declared
mgroover@knuth:~/compilers/lab4>
```

If I cannot declare a variable more than once

```
mgroover@knuth:~/compilers/lab4> ./lab4
int y;
Whitespace found
Letter found
return a token ;
cariage return

ya =y;
Letter found
Whitespace found
return a token =
Letter found
return a token ;
the variable ya doesn't exists
mgroover@knuth:~/compilers/lab4>
```

I can't use a variable that is not declared

```
mgroover@knuth:~/compilers/lab4> ./lab4
```

```
int x;
```

```
Whitespace found
```

```
Letter found
```

```
return a token ;
```

```
cariage return
```

```
x=4
```

```
Letter found
```

```
return a token =
```

```
Digit found
```

```
found an integer
```

```
cariage return
```

```
the variable x
```

```
is now 4
```

```
x= x*4
```

```
Letter found
```

```
return a token =
```

```
Whitespace found
```

```
Letter found
```

```
return a token *
```

```
Digit found
```

```
found an integer
```

```
cariage return
```

```
the variable x
```

```
is now 16
```

Showing that the variable can be used on both the left and right hand side