

## CD Lab Assignment 7

Name : Nalanaga Ysaswin

Roll no : 20CS01040

Instructions to compile :

Go to the directory and

- 1) yacc parser.y -d // to generate a header file
- 2) flex 20CS01040.l // to generate a lex.yy.c file
- 3) gcc y.tab.c // to generate a.out file
- 4) ./a.out

Sample input :

global

```
def
    // some type defs
    someId := product
        pr1op[3], pr2op : someType;
        otherProp123 :
        otherId := product
            extra, nul : null
        end
    end;

    var1, var2[5], var3 : float;
    var4[1], v5[3][4] : int
end

fun myFunc -> int
    print "# 123QWE"
end
```

```
fun myFunc2
```

```
    var1, var2[5], var3 : float;
```

```
    var4[1], v5[3][4] : int
```

```
    -> int
```

```
    print "# 123QWE"
```

```
end
```

```
// Some comments in between
```

```
// int print scan
```

```
while awe > qwe:
```

```
    print "# 123QWE";
```

```
    awe := qwe + 1;
```

```
    qwe := qwe + 2
```

```
end;
```

```
if awe <> qwe or awe <> qwe or qwe + abc > 50 and not((abc = 10.234) or 10.234 <= qwe and not  
qwe > 50) or qwe <> abc:
```

```
    print "# 123QWE"
```

```
else
```

```
    print "ABC";
```

```
    print "ABC"
```

```
end;
```

```
from iteratorId := 5+abc to qwe-1.23 step abc+qwe :
```

```
    print "ABC";
```

```
    print "Looping"
```

```
end;
```

```
exit;
```

```
return someExp1 mod someExp2;
```

```
super.current.sub := (id1 : 1+2, 3-4, 5*6, 7/8, 9 mod 10, +1.-2, 4.567, (3), id2) ;  
print "ABC";  
read %f abc + 10.234 mod qwe / 50;  
  
read %s someID0 [id1 + id2][id3 - id4]
```

end

## The output:

parsed successfully

the given program is syntactically and lexically accurate