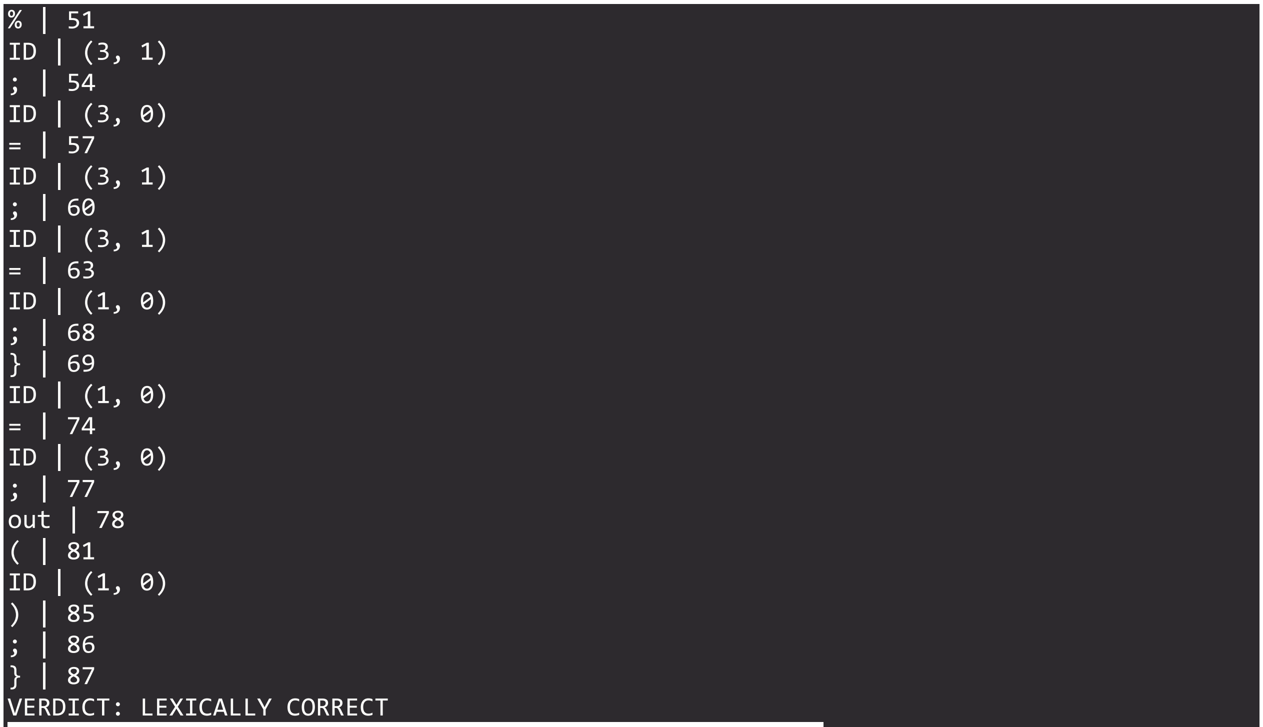
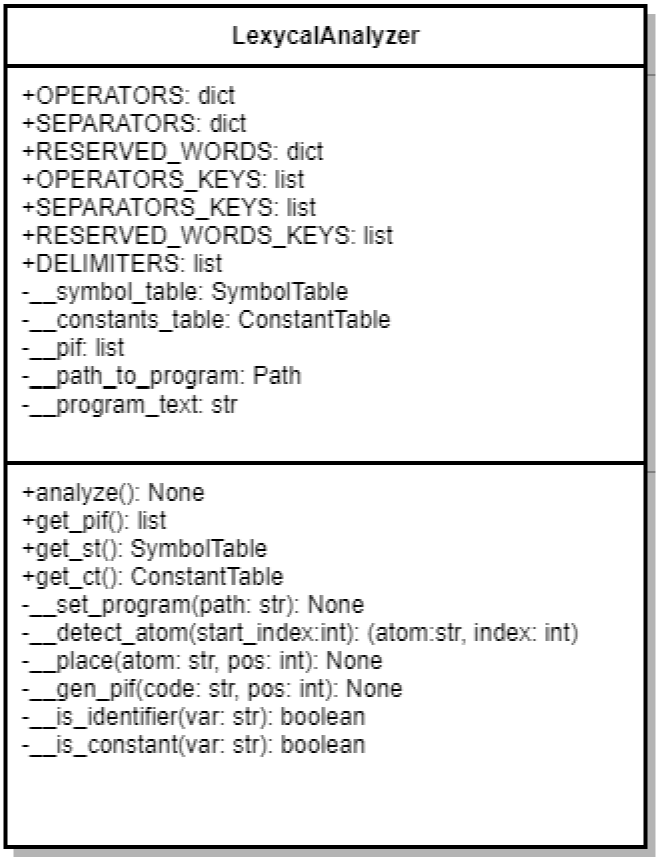
```
https://github.com/banuAndrei99/FLCD


# Lexical Analyzer
## __set_program(program: string)
    Reads the text content of the program and formats it.
   - pre: program should be a valid path
   - post: attribute __program_text is set

## get_pif()
   Returns the pif
   - pre: None
   - post: None

## get_st()
   Returns the SymbolTable object
   - pre: None
   - post None

## get_ct()
    Returns the ConstantsTable object
    - pre: None
    - post: None

## analyze():
    Analyzes the program, filling up the ST, CT and PIF.
    - pre: __set_program() was called
    - post: PIF, ST and CT contain the data from the program

### __detect_atom(start_index: 0):
    Returns the next atom and its end index. The search begins from `start_index`
    - pre: __set_program() was called
    - post: None

### __place(atom: str, pos: int):
    Places the atom in st or ct if needed and adds a new entry in pif.
    Raises error in case syntax issues
    - pre: None
    - post: adds a new entry to __pif. Can also add new entries to __symbol_table and
__constant_table.

#### __is_identifier(var: string):
    Checks if var is a valid identifier.
    - pre: None
    - post: None

#### __is_constant(var: string):
    Checks if var is a valid constant.
    - pre: None
    - post: None

#### __gen_pif(code: str, pos: int | tuple):
    Adds a new entry in pif.
    - pre: None
    - post: pif is modified or error is raised.
```

~~~
## Input example:
```python
main{
   int a = 36, b = 30, gcd;
   while(b != 0){
      gcd = a % b;
      a = b;
      b = gcd;
   }
   gcd = a;
   out(gcd);
}
```
## Output:
ANALYZING p2
Symbol Table:
0 --> []
1 --> [gcd,]
2 --> []
3 --> [a,b,]
4 --> []
5 --> []
6 --> []
Constant Table:
0 --> []
1 --> []
2 --> []
3 --> []
4 --> [36,0,]
5 --> [30,]
6 --> []
PIF:
main | 0
{ | 4
int | 5
ID | (3, 0)
= | 11
CONST | (4, 0)
, | 15
ID | (3, 1)
= | 19
CONST | (5, 0)
, | 23
ID | (1, 0)
; | 28
while | 29
( | 34
ID | (3, 1)
!= | 37
CONST | (4, 1)
) | 41
{ | 42
ID | (1, 0)
= | 47
ID | (3, 0)
~~~



```
% | 51
ID | (3, 1)
; | 54
ID | (3, 0)
= | 57
ID | (3, 1)
; | 60
ID | (3, 1)
= | 63
ID | (1, 0)
; | 68
} | 69
ID | (1, 0)
= | 74
ID | (3, 0)
; | 77
out | 78
( | 81
ID | (1, 0)
) | 85
; | 86
} | 87
VERDICT: LEXICALLY CORRECT
```



```
LexycalAnalyzer

+OPERATORS: dict
+SEPARATORS: dict
+RESERVED_WORDS: dict
+OPERATORS_KEYS: list
+SEPARATORS_KEYS: list
+RESERVED_WORDS_KEYS: list
+DELIMITERS: list
-__symbol_table: SymbolTable
-__constants_table: ConstantTable
-__pif: list
-__path_to_program: Path
-__program_text: str

+analyze(): None
+get_pif(): list
+get_st(): SymbolTable
+get_ct(): ConstantTable
-__set_program(path: str): None
-__detect_atom(start_index:int): (atom:str, index: int)
-__place(atom: str, pos: int): None
-__gen_pif(code: str, pos: int): None
-__is_identifier(var: str): boolean
-__is_constant(var: str): boolean
```

```
More examples can be found on github.
```