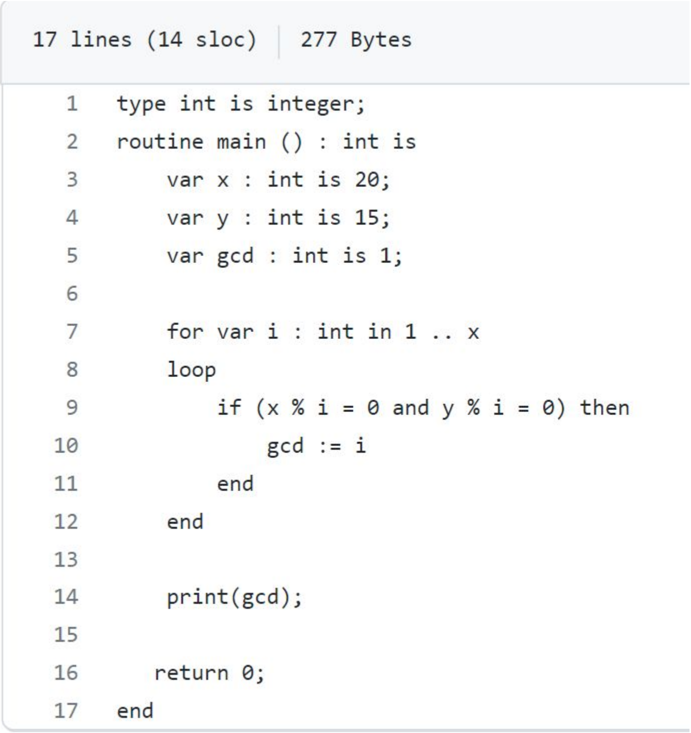
Carbonic-C lexer

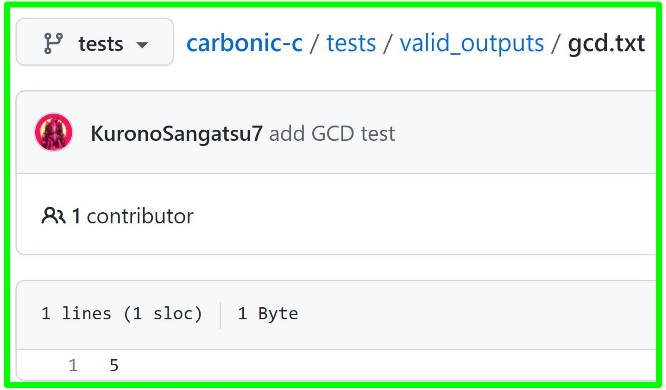
Asem Abdelhady
Jaffar Totanji
Menna Awadallah
Mosab Mohamed

GCD in Carbonic-C



17 lines (14 sloc) | 277 Bytes

```
type int is integer;
routine main () : int is
    var x : int is 20;
    var y : int is 15;
    var gcd : int is 1;

    for var i : int in 1 .. x
    loop
        if (x % i = 0 and y % i = 0) then
            gcd := i
        end
    end

    print(gcd);

   return 0;
end
```



carbonic-c / tests / valid_outputs / gcd.txt

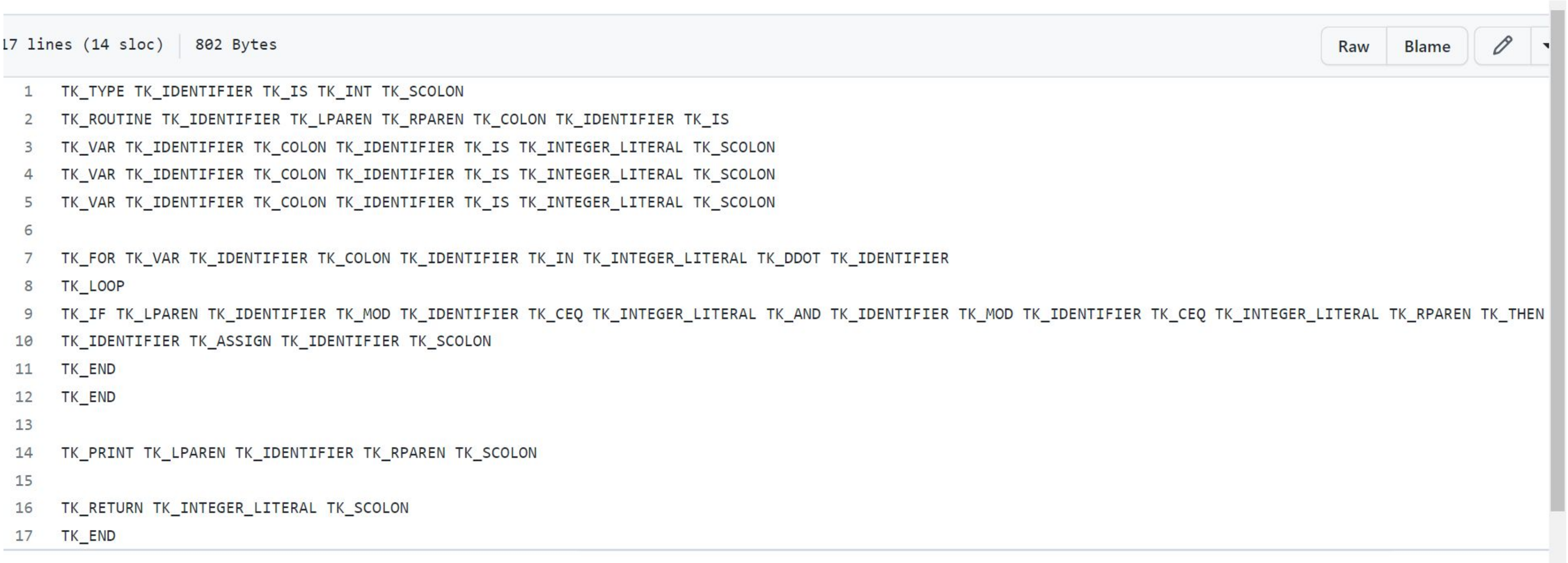
KuronoSangatsu7 add GCD test

1 contributor

1 lines (1 sloc) | 1 Byte

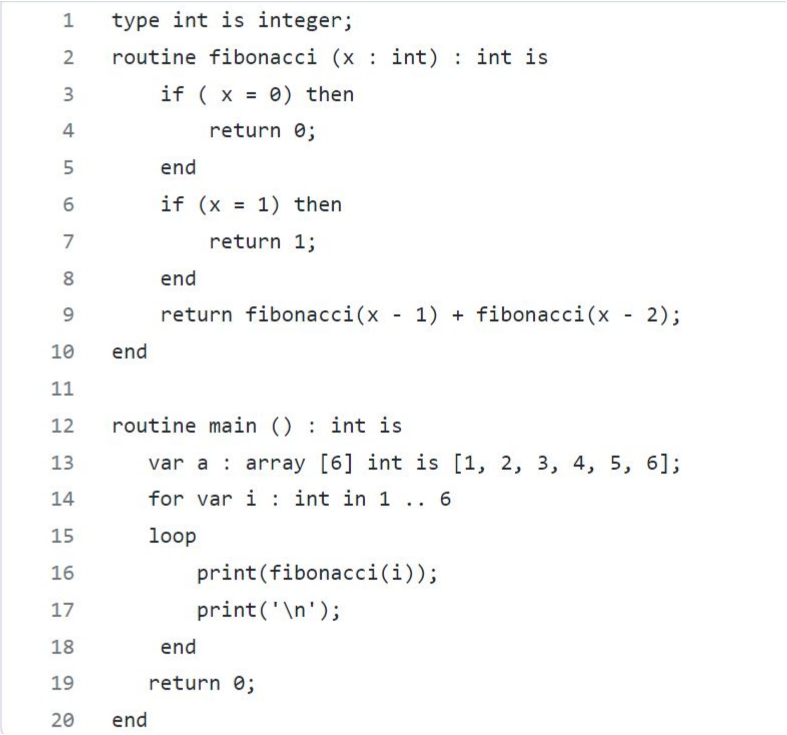
```
5
```

GCD’s tokens

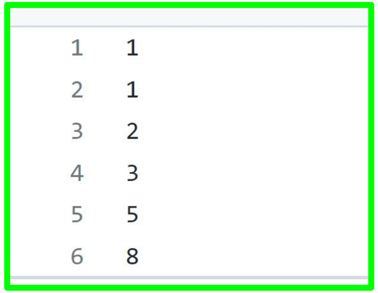


```
TK_TYPE TK_IDENTIFIER TK_IS TK_INT TK_SCOLON
TK_ROUTINE TK_IDENTIFIER TK_LPAREN TK_RPAREN TK_COLON TK_IDENTIFIER TK_IS
TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IS TK_INTEGER_LITERAL TK_SCOLON
TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IS TK_INTEGER_LITERAL TK_SCOLON
TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IS TK_INTEGER_LITERAL TK_SCOLON

TK_FOR TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IN TK_INTEGER_LITERAL TK_DDOT TK_IDENTIFIER
TK_LOOP
TK_IF TK_LPAREN TK_IDENTIFIER TK_MOD TK_IDENTIFIER TK_CEQ TK_INTEGER_LITERAL TK_AND TK_IDENTIFIER TK_MOD TK_IDENTIFIER TK_CEQ TK_INTEGER_LITERAL TK_RPAREN TK_THEN
TK_IDENTIFIER TK_ASSIGN TK_IDENTIFIER TK_SCOLON
TK_END
TK_END

TK_PRINT TK_LPAREN TK_IDENTIFIER TK_RPAREN TK_SCOLON

TK_RETURN TK_INTEGER_LITERAL TK_SCOLON
TK_END
```

Fibonacci in Carbonic-C

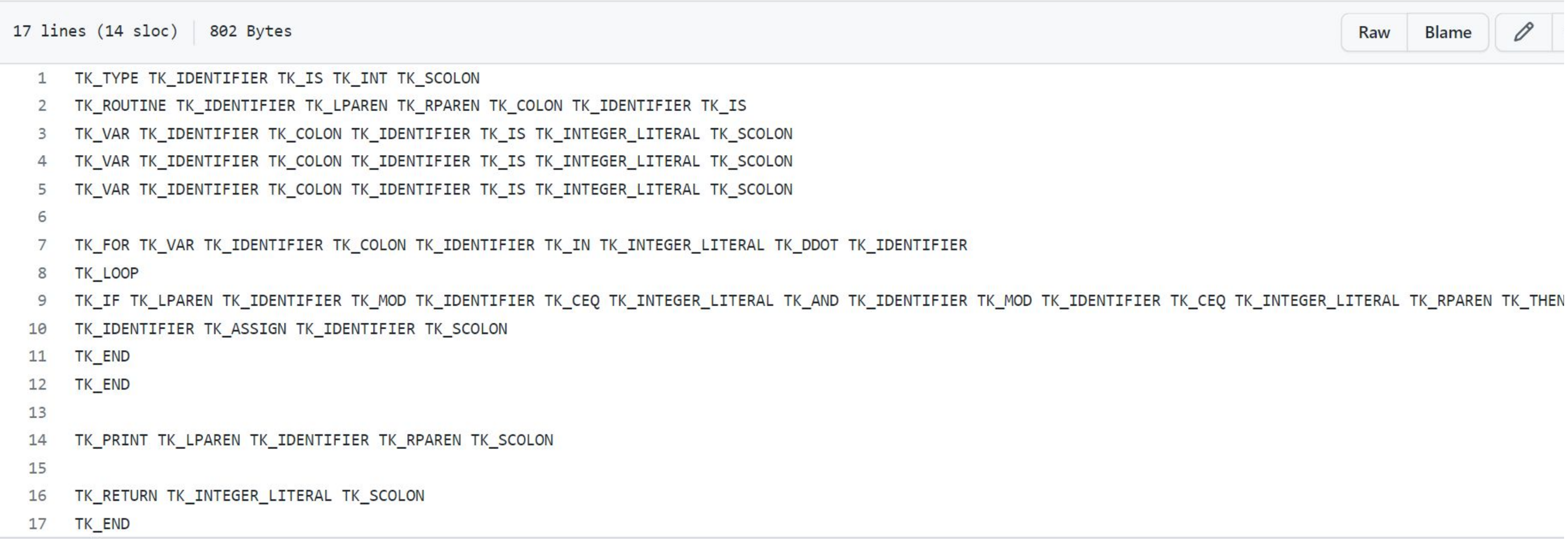


```
type int is integer;
routine fibonacci (x : int) : int is
    if ( x = 0) then
        return 0;
    end
    if (x = 1) then
        return 1;
    end
    return fibonacci(x - 1) + fibonacci(x - 2);
end

routine main () : int is
   var a : array [6] int is [1, 2, 3, 4, 5, 6];
   for var i : int in 1 .. 6
   loop
       print(fibonacci(i));
       print('\n');
    end
   return 0;
end
```



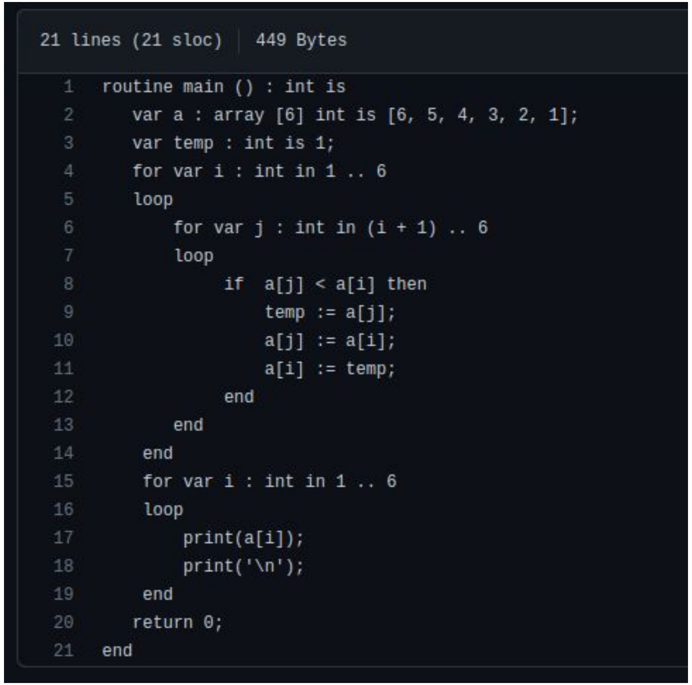
```
1
1
2
3
5
8
```

Fibonacci’s tokens



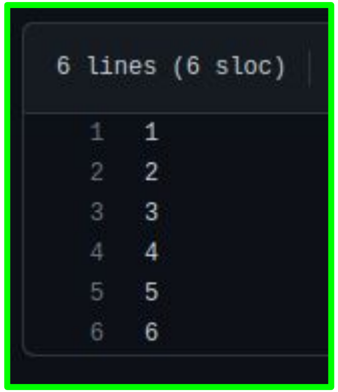
```
TK_TYPE TK_IDENTIFIER TK_IS TK_INT TK_SCOLON
TK_ROUTINE TK_IDENTIFIER TK_LPAREN TK_RPAREN TK_COLON TK_IDENTIFIER TK_IS
TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IS TK_INTEGER_LITERAL TK_SCOLON
TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IS TK_INTEGER_LITERAL TK_SCOLON
TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IS TK_INTEGER_LITERAL TK_SCOLON

TK_FOR TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IN TK_INTEGER_LITERAL TK_DDOT TK_IDENTIFIER
TK_LOOP
TK_IF TK_LPAREN TK_IDENTIFIER TK_MOD TK_IDENTIFIER TK_CEQ TK_INTEGER_LITERAL TK_AND TK_IDENTIFIER TK_MOD TK_IDENTIFIER TK_CEQ TK_INTEGER_LITERAL TK_RPAREN TK_THEN
TK_IDENTIFIER TK_ASSIGN TK_IDENTIFIER TK_SCOLON
TK_END
TK_END

TK_PRINT TK_LPAREN TK_IDENTIFIER TK_RPAREN TK_SCOLON

TK_RETURN TK_INTEGER_LITERAL TK_SCOLON
TK_END
```

Bubble sort in Carbonic-C



21 lines (21 sloc) | 449 Bytes

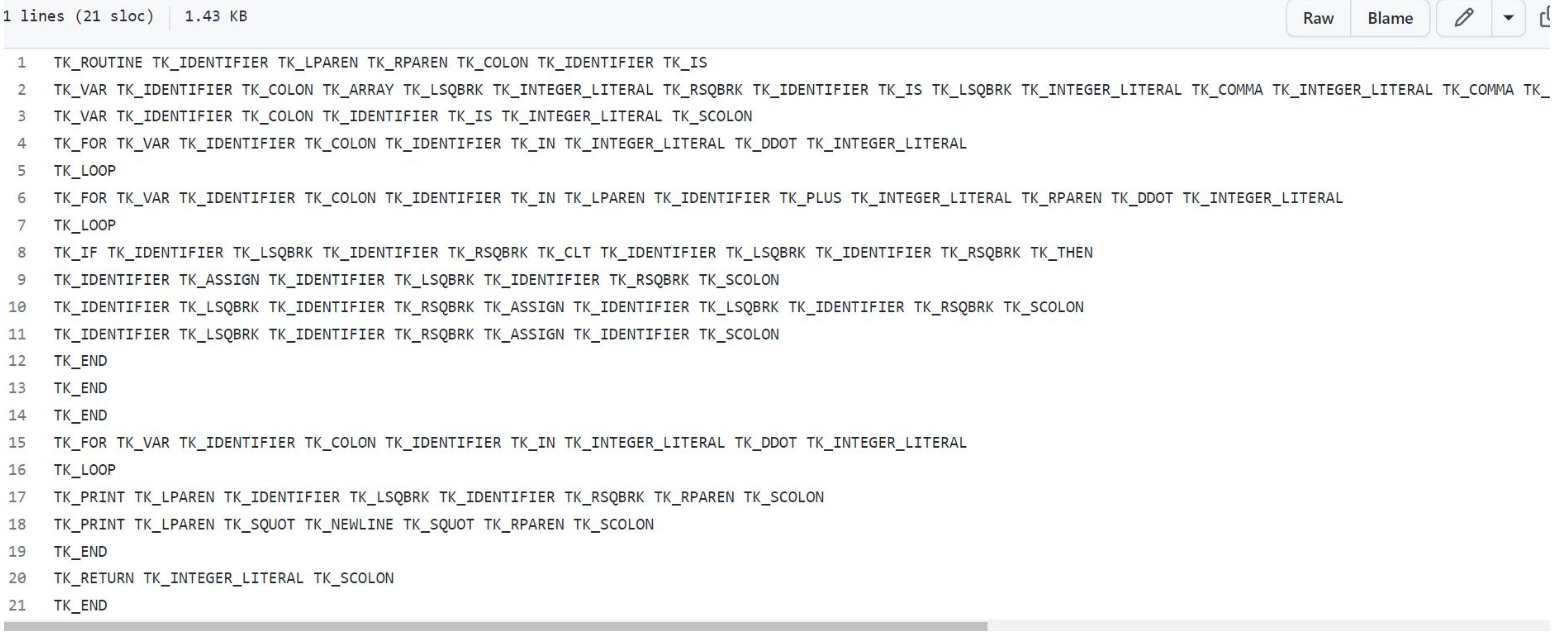
```
routine main () : int is
   var a : array [6] int is [6, 5, 4, 3, 2, 1];
   var temp : int is 1;
   for var i : int in 1 .. 6
   loop
       for var j : int in (i + 1) .. 6
       loop
            if  a[j] < a[i] then
                temp := a[j];
                a[j] := a[i];
                a[i] := temp;
            end
       end
    end
    for var i : int in 1 .. 6
    loop
        print(a[i]);
        print('\n');
    end
   return 0;
end
```



6 lines (6 sloc)

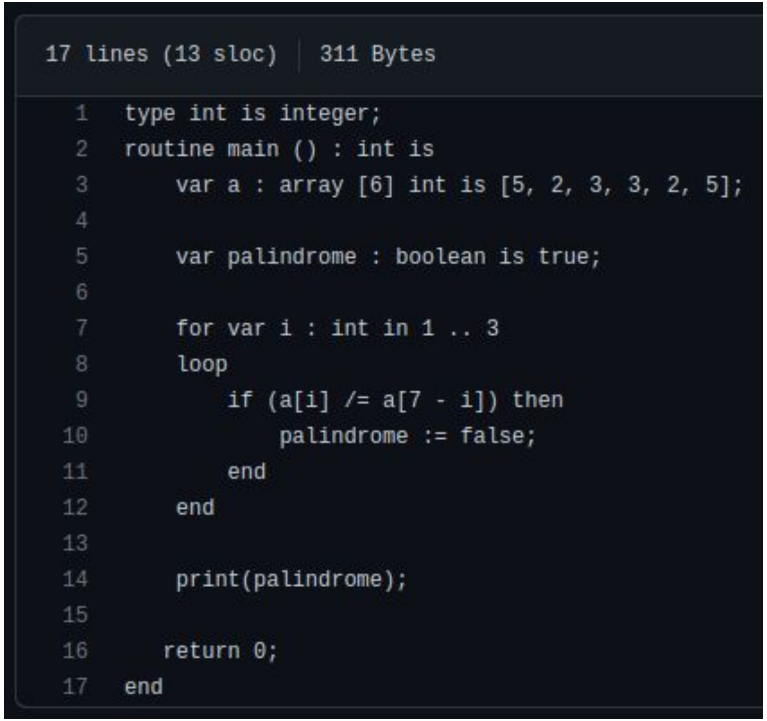
```
1
2
3
4
5
6
```

Bubble sort's tokens



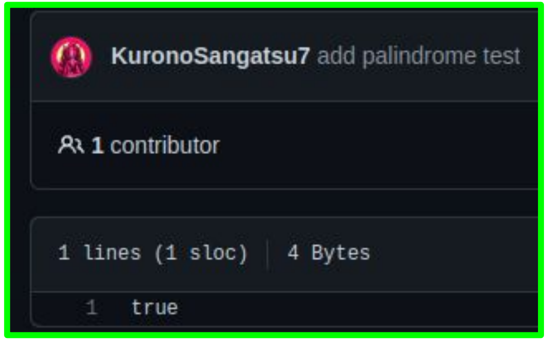
```
TK_ROUTINE TK_IDENTIFIER TK_LPAREN TK_RPAREN TK_COLON TK_IDENTIFIER TK_IS
TK_VAR TK_IDENTIFIER TK_COLON TK_ARRAY TK_LSQBRK TK_INTEGER_LITERAL TK_RSQBRK TK_IDENTIFIER TK_IS TK_LSQBRK TK_INTEGER_LITERAL TK_COMMA TK_INTEGER_LITERAL TK_COMMA TK_
TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IS TK_INTEGER_LITERAL TK_SCOLON
TK_FOR TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IN TK_INTEGER_LITERAL TK_DDOT TK_INTEGER_LITERAL
TK_LOOP
TK_FOR TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IN TK_LPAREN TK_IDENTIFIER TK_PLUS TK_INTEGER_LITERAL TK_RPAREN TK_DDOT TK_INTEGER_LITERAL
TK_LOOP
TK_IF TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_CLT TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_THEN
TK_IDENTIFIER TK_ASSIGN TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_SCOLON
TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_ASSIGN TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_SCOLON
TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_ASSIGN TK_IDENTIFIER TK_SCOLON
TK_END
TK_END
TK_END
TK_FOR TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IN TK_INTEGER_LITERAL TK_DDOT TK_INTEGER_LITERAL
TK_LOOP
TK_PRINT TK_LPAREN TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_RPAREN TK_SCOLON
TK_PRINT TK_LPAREN TK_SQUOT TK_NEWLINE TK_SQUOT TK_RPAREN TK_SCOLON
TK_END
TK_RETURN TK_INTEGER_LITERAL TK_SCOLON
TK_END
```

Palindrome checking in Carbonic-C



17 lines (13 sloc) | 311 Bytes

```
type int is integer;
routine main () : int is
    var a : array [6] int is [5, 2, 3, 3, 2, 5];

    var palindrome : boolean is true;

    for var i : int in 1 .. 3
    loop
        if (a[i] /= a[7 - i]) then
            palindrome := false;
        end
    end

    print(palindrome);

   return 0;
end
```



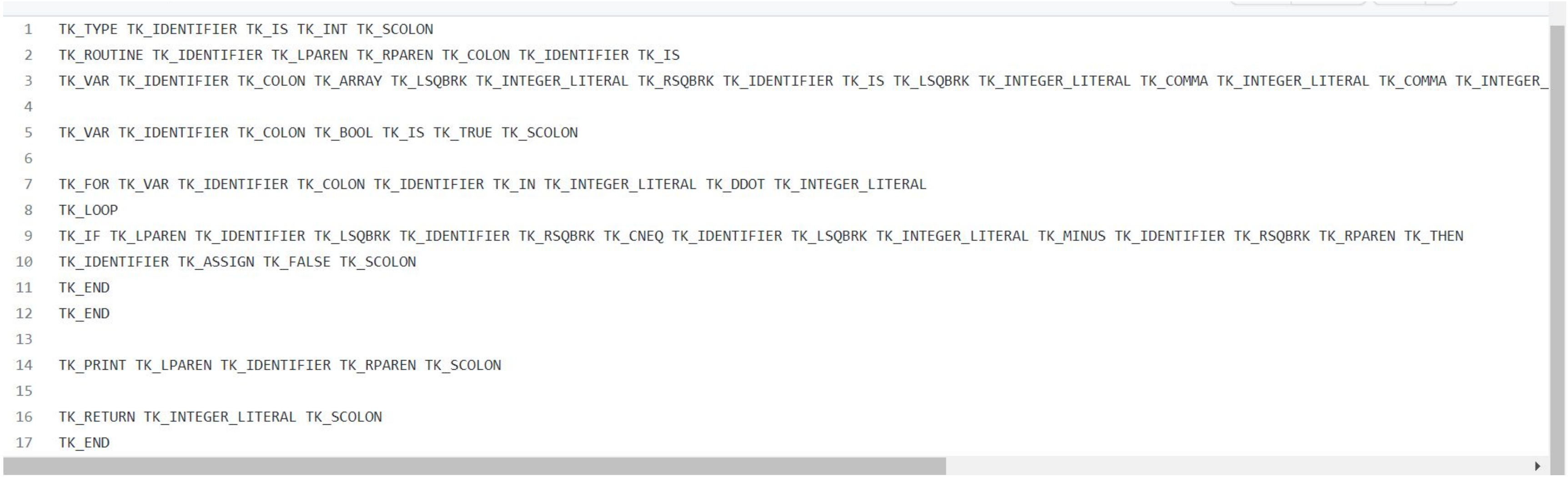
KuronoSangatsu7 add palindrome test

1 contributor

1 lines (1 sloc) | 4 Bytes

```
true
```

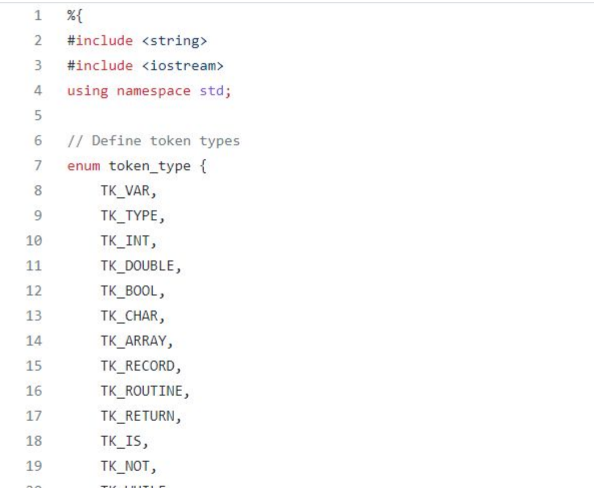
Palindrome Checking’s tokens



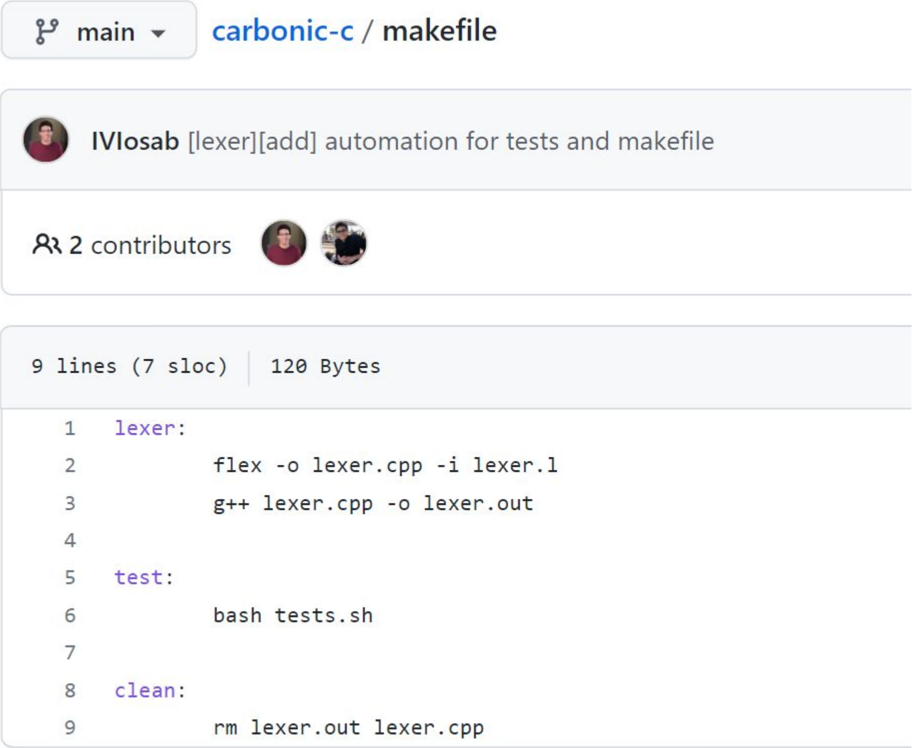
```
TK_TYPE TK_IDENTIFIER TK_IS TK_INT TK_SCOLON
TK_ROUTINE TK_IDENTIFIER TK_LPAREN TK_RPAREN TK_COLON TK_IDENTIFIER TK_IS
TK_VAR TK_IDENTIFIER TK_COLON TK_ARRAY TK_LSQBRK TK_INTEGER_LITERAL TK_RSQBRK TK_IDENTIFIER TK_IS TK_LSQBRK TK_INTEGER_LITERAL TK_COMMA TK_INTEGER_LITERAL TK_COMMA TK_INTEGER_

TK_VAR TK_IDENTIFIER TK_COLON TK_BOOL TK_IS TK_TRUE TK_SCOLON

TK_FOR TK_VAR TK_IDENTIFIER TK_COLON TK_IDENTIFIER TK_IN TK_INTEGER_LITERAL TK_DDOT TK_INTEGER_LITERAL
TK_LOOP
TK_IF TK_LPAREN TK_IDENTIFIER TK_LSQBRK TK_IDENTIFIER TK_RSQBRK TK_CNEQ TK_IDENTIFIER TK_LSQBRK TK_INTEGER_LITERAL TK_MINUS TK_IDENTIFIER TK_RSQBRK TK_RPAREN TK_THEN
TK_IDENTIFIER TK_ASSIGN TK_FALSE TK_SCOLON
TK_END
TK_END

TK_PRINT TK_LPAREN TK_IDENTIFIER TK_RPAREN TK_SCOLON

TK_RETURN TK_INTEGER_LITERAL TK_SCOLON
TK_END
```

Implementation details

- Flex for lexer
- MakeFile for defining compilation
- Bash script for automation of tests



```
%{
#include <string>
#include <iostream>
using namespace std;

// Define token types
enum token_type {
    TK_VAR,
    TK_TYPE,
    TK_INT,
    TK_DOUBLE,
    TK_BOOL,
    TK_CHAR,
    TK_ARRAY,
    TK_RECORD,
    TK_ROUTINE,
    TK_RETURN,
    TK_IS,
    TK_NOT,
    TK_WHILE
```



carbonic-c / makefile

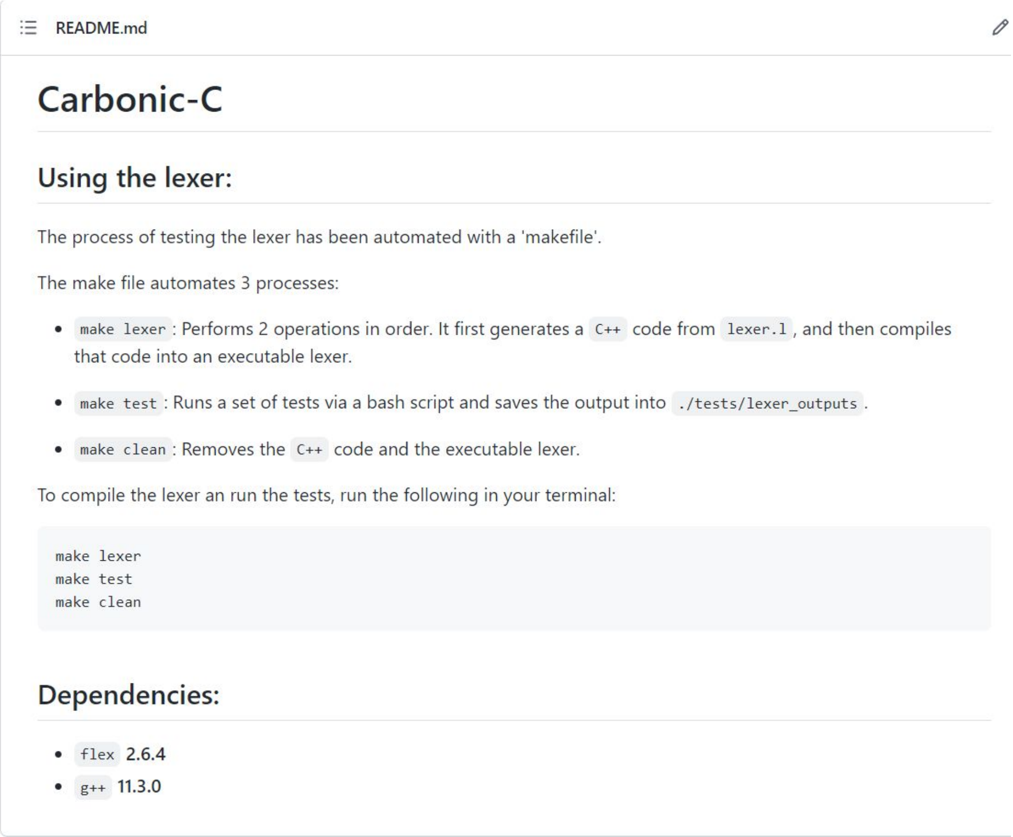
IVlosab [lexer][add] automation for tests and makefile

2 contributors

9 lines (7 sloc) | 120 Bytes

```
lexer:
        flex -o lexer.cpp -i lexer.l
        g++ lexer.cpp -o lexer.out

test:
        bash tests.sh

clean:
        rm lexer.out lexer.cpp
```

Read me



Carbonic-C

Using the lexer:

The process of testing the lexer has been automated with a 'makefile'.

The make file automates 3 processes:

- `make lexer` : Performs 2 operations in order. It first generates a `C++` code from `lexer.l`, and then compiles that code into an executable lexer.
- `make test` : Runs a set of tests via a bash script and saves the output into `./tests/lexer_outputs` .
- `make clean` : Removes the `C++` code and the executable lexer.

To compile the lexer an run the tests, run the following in your terminal:

```
make lexer
make test
make clean
```

Dependencies:

- `flex` 2.6.4
- `g++` 11.3.0

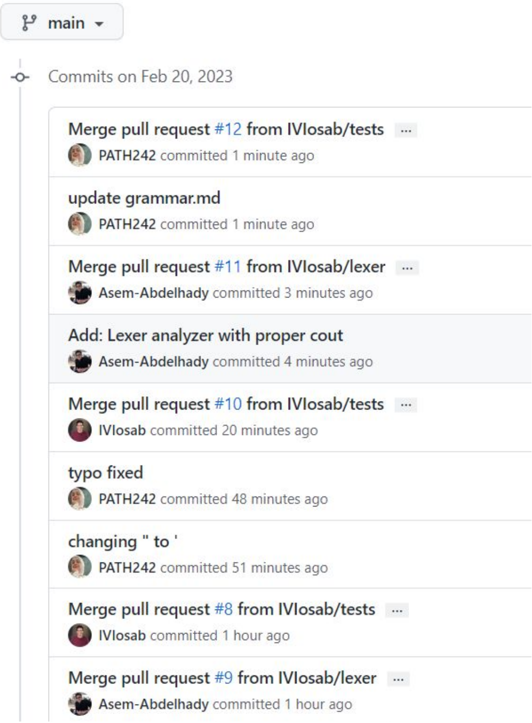
Team responsibilities

Asem: Lexer

Jaffar: Tests + documentation

Menna : Tests + formal grammar definitions

Mosab: lexer debugging, automation of testing



main

Commits on Feb 20, 2023

Merge pull request #12 from IVIosab/tests ···
PATH242 committed 1 minute ago

update grammar.md
PATH242 committed 1 minute ago

Merge pull request #11 from IVIosab/lexer ···
Asem-Abdelhady committed 3 minutes ago

Add: Lexer analyzer with proper cout
Asem-Abdelhady committed 4 minutes ago

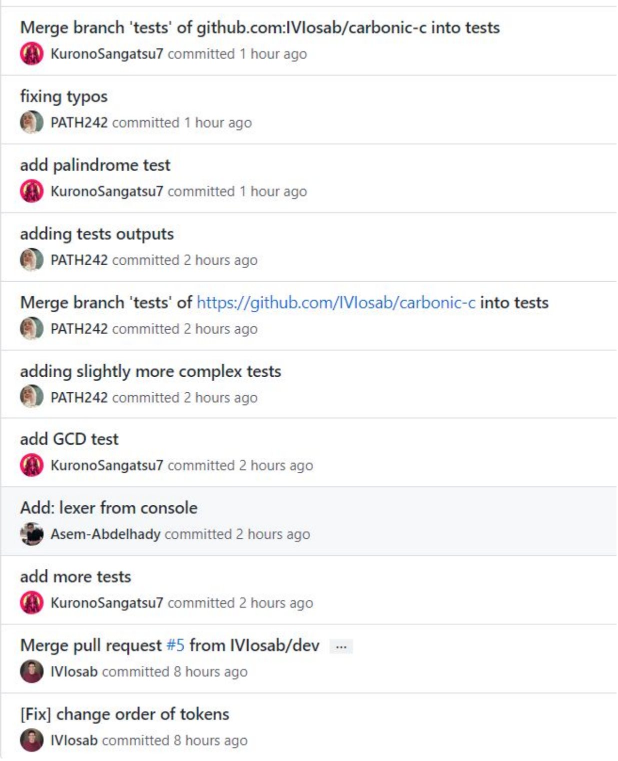
Merge pull request #10 from IVIosab/tests ···
IVIosab committed 20 minutes ago

typo fixed
PATH242 committed 48 minutes ago

changing " to '
PATH242 committed 51 minutes ago

Merge pull request #8 from IVIosab/tests ···
IVIosab committed 1 hour ago

Merge pull request #9 from IVIosab/lexer ···
Asem-Abdelhady committed 1 hour ago



Merge branch 'tests' of github.com:IVIosab/carbonic-c into tests
KuronoSangatsu7 committed 1 hour ago

fixing typos
PATH242 committed 1 hour ago

add palindrome test
KuronoSangatsu7 committed 1 hour ago

adding tests outputs
PATH242 committed 2 hours ago

Merge branch 'tests' of https://github.com/IVIosab/carbonic-c into tests
PATH242 committed 2 hours ago

adding slightly more complex tests
PATH242 committed 2 hours ago

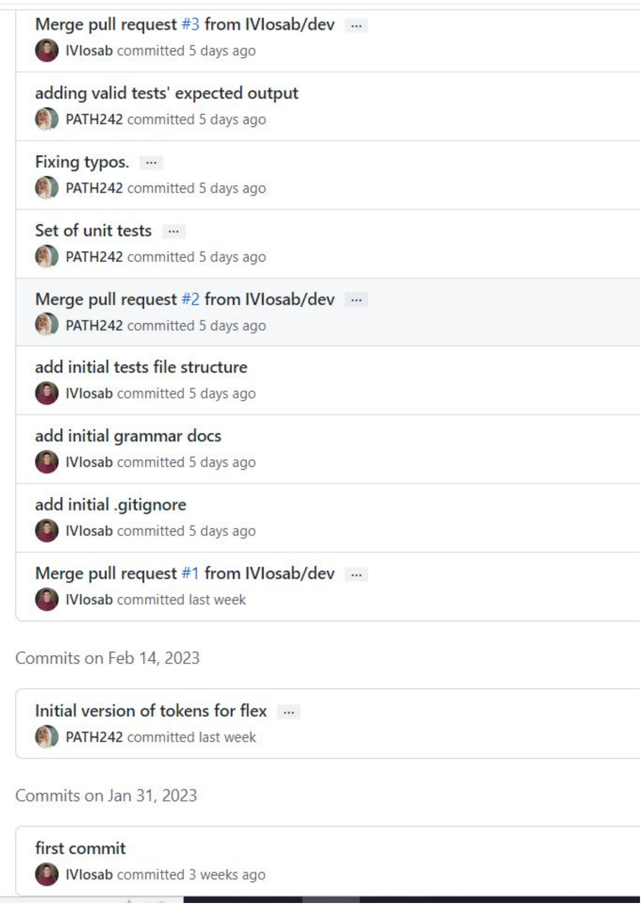
add GCD test
KuronoSangatsu7 committed 2 hours ago

Add: lexer from console
Asem-Abdelhady committed 2 hours ago

add more tests
KuronoSangatsu7 committed 2 hours ago

Merge pull request #5 from IVIosab/dev ···
IVIosab committed 8 hours ago

[Fix] change order of tokens
IVIosab committed 8 hours ago



Merge pull request #3 from IVIosab/dev ···
IVIosab committed 5 days ago

adding valid tests' expected output
PATH242 committed 5 days ago

Fixing typos. ···
PATH242 committed 5 days ago

Set of unit tests ···
PATH242 committed 5 days ago

Merge pull request #2 from IVIosab/dev ···
PATH242 committed 5 days ago

add initial tests file structure
IVIosab committed 5 days ago

add initial grammar docs
IVIosab committed 5 days ago

add initial .gitignore
IVIosab committed 5 days ago

Merge pull request #1 from IVIosab/dev ···
IVIosab committed last week

Commits on Feb 14, 2023

Initial version of tokens for flex ···
PATH242 committed last week

Commits on Jan 31, 2023

first commit
IVIosab committed 3 weeks ago

Please see our code below

https://github.com/lVlosab/carbonic-c

Now for demo!

Thank you!