

CSE 351 PROJECT

A lex yacc program that will convert a python code to c++ code. The input python codes can contain only assignment and if/else statements. The project is to convert the given assignment and if/else statements into c++ language.

20200702023

Elif Çolak

First, I coded the syntax analysis part, then the semantic analysis part. Semantic analysis part can catch tab inconsistency, if/else inconsistency and type insistency. Lastly, I converted given python code to c++ code and print the code inside the “**tocpp:**”.

I controlled my outputs for given inputs like this:

./project input1.txt > outputStudent.txt

diff -s output1.txt outputStudent.txt

and for all input files I can successfully do syntax and semantic analysis.

Some of the samples from my analysis:

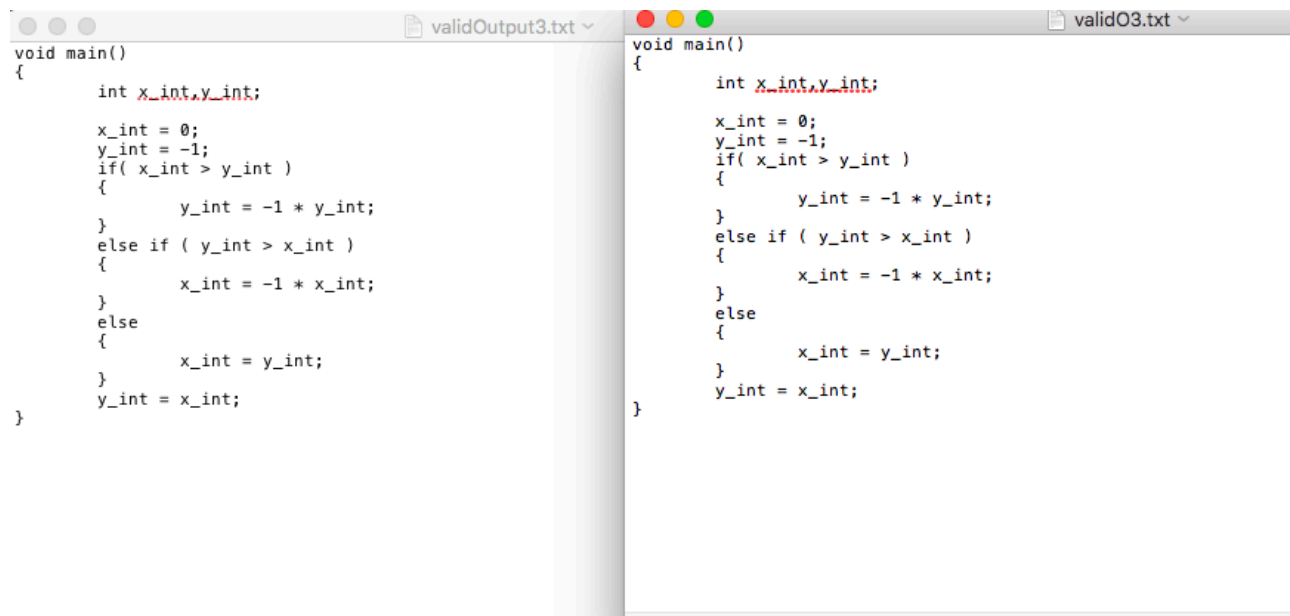
```
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % make
clang: warning: treating 'c' input as 'c++' when in C++ mode, this behavior is deprecated [-Wdeprecated]
clang: warning: treating 'c' input as 'c++' when in C++ mode, this behavior is deprecated [-Wdeprecated]
ld: warning: object file (/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/
Ks/MacOSX.sdk/usr/lib/libl.a[arm64][3](libyywrap.o)) was built for newer 'macOS' version (14.2) than be
linked (14.0)
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid1.txt > invalid01.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput1.txt invalid01.txt
Files invalidOutput1.txt and invalid01.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid2.txt > invalid02.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput2.txt invalid02.txt
Files invalidOutput2.txt and invalid02.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid3.txt > invalid03.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput3.txt invalid03.txt
Files invalidOutput3.txt and invalid03.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid4.txt > invalid04.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput4.txt invalid04.txt
Files invalidOutput4.txt and invalid04.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid5.txt > invalid05.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput5.txt invalid05.txt
Files invalidOutput5.txt and invalid05.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid6.txt > invalid06.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput6.txt invalid06.txt
Files invalidOutput6.txt and invalid06.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid7.txt > invalid07.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput7.txt invalid07.txt
Files invalidOutput7.txt and invalid07.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project invalid8.txt > invalid08.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s invalidOutput8.txt invalid08.txt
Files invalidOutput8.txt and invalid08.txt are identical
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % ./project valid1.txt > valid01.txt
elifcolak@Elifts-MacBook-Air project_CSE351_20200702023 % diff -s validOutput1.txt valid01.txt
```

valid3.txt

```
$ ./project valid3.txt > valid03.txt
```

```
$ diff -s validOutput3.txt valid03.txt
```

Files validOutput3.txt and valid03.txt are identical



The image shows two side-by-side code editors. The left editor is titled 'validOutput3.txt' and the right editor is titled 'valid03.txt'. Both editors contain the same C code:

```
void main()
{
    int x_int,y_int;

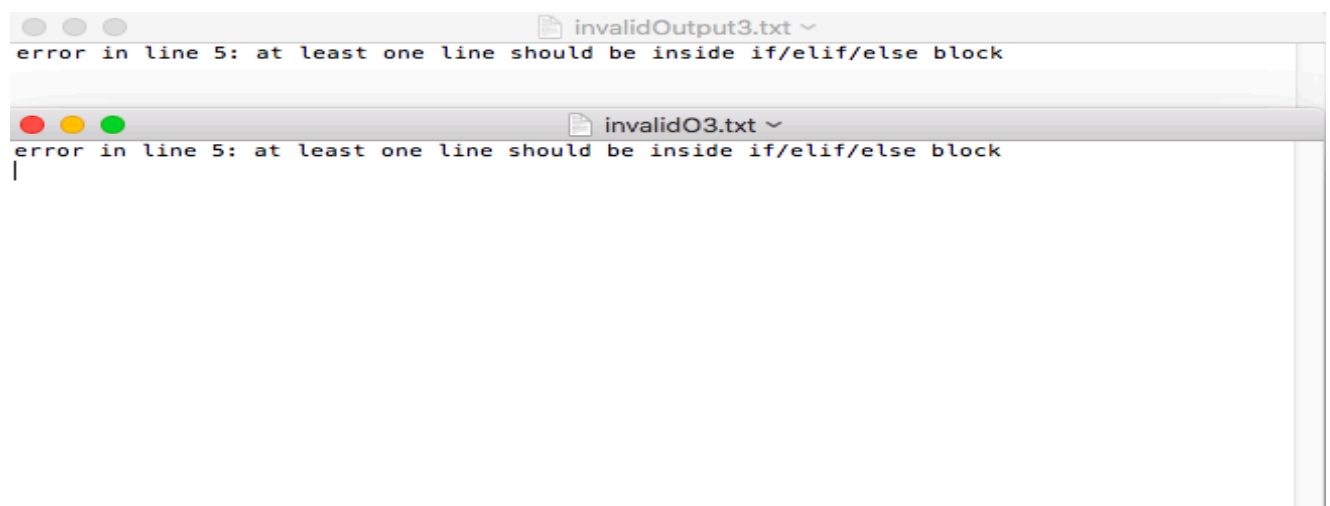
    x_int = 0;
    y_int = -1;
    if( x_int > y_int )
    {
        y_int = -1 * y_int;
    }
    else if ( y_int > x_int )
    {
        x_int = -1 * x_int;
    }
    else
    {
        x_int = y_int;
    }
    y_int = x_int;
}
```

Invalid3.txt

```
./project invalid3.txt > invalid03.txt
```

```
diff -s invalidOutput3.txt invalid03.txt
```

Files invalidOutput3.txt and invalid03.txt are identical



The image shows two side-by-side code editors. The left editor is titled 'invalidOutput3.txt' and the right editor is titled 'invalid03.txt'. Both editors contain the same C code as in the previous example, but with a compiler error message displayed below the code:

```
error in line 5: at least one line should be inside if/elif/else block
```