

Phase 5 implementation

Author: Bruno Liu fc56297

For the phase 5 I decided to implement the imports, so that the user can import other functions to use in their program.

The syntax for the imports are:

```
from <path_to_file> import *;
```

This will import all the functions and global variables from the file except the main function.

Or the syntax can be:

```
from <path_to_file> import <list_of_function_name_separated_by_commas>;
```

This will import only the functions from the file that are in the list. If the any function is not in the file an error will be raised.

An example of the implementation is in the exampleForImports folder, where the main.pl file imports everything from second.pl and the second.pl imports the fibonacci function from third.pl. Consequently, the main.pl can use the fibonacci function from the third.pl file.

The main.pl has:

```
from "./exampleForImports/second.pl" import *;
```

The second.pl has:

```
from "./exampleForImports/third.pl" import fibonacci;
```

This implementation was done in the plush_compiler.py file, between the first parsing and the semantic analysis part, where the imports are resolved after the parsing, by check if there is any import in the ast, and if there is, it is done the parsing of the file that is being imported and the ast of the imported file is added to the ast of the main file. This is done recursively, so that if the imported file has any import, it is also resolved.

Limitations:

When importing specific functions, the dependencies of that specific function are not imported, so if the function that is being imported has any dependencies, they must be also declared.