Progress Report A8

Vision: We are creating a python interpreter in OCaml, called OPython. This involves a text user interface, a python parser, some kind of evaluation, assigning values to variables, return statements and for and while loops. We may include functions and objects if time permits in later sprints.

Summary of Progress: We implemented for-loops. We fixed the while loops and some built-in functions such as array slicing. We implemented Python dictionaries, along with getting and putting new keys and values. We created tests in .ml as well as in .py, which can be run in OPython. Finally, we implemented list comprehensions as well.

Activity Breakdown:

Eric:

1. Built-in functions: max, min - both take any args
2. Implemented quit function
3. Testing line inputs, parsing, evaluating with given states
4. Accommodating new fields into cross-typing helpers
5. Documentation for built-in functions

Patrick:

1. Parsing nested lists and slicing nested lists.
2. Parsing Dictionaries and getting values from keys
3. Parsing += and similar operations in more situations
4. Parsing list comprehensions
5. Formatting and documentations

William:

1. Python Dictionaries

2. Dictionary functions: put and get
3. Documentation and testing (in test.ml)

Zaibo:

1. Add parsing and evaluation of for loop
2. Add parsing and evaluation of list comprehensions
3. Create testing framework to detect errors in code
4. Add tests by implementing (and copying) classic algorithms for testing

Productivity Analysis:

We finished all the code involving dictionaries, for loops, list comprehensions, and additional built-in functions on Monday and refined the code leading up to Wednesday. We were more proactive this week than ever, making sure our code was robust with extensive testing in both Python and OCaml files. We created several demo functions to show the new features like for loops and dictionaries. These can be run using `make pytest`. We created a quit function and added asserts. Overall, this week was adding distinctive Python features and polishing our code.

Scope Grade:

Excellent: For loops, List comprehension, dictionaries, some more built-in functions, documentation, and testing

Satisfactory: Dictionaries, documentation, testing

Unsatisfactory: Not doing dictionaries

We grade ourselves Excellent because we did everything that is part of the scope. We were also able support recursion.

Goals for Next Sprint: This is the final sprint.

Coding standards grades:

Comprehensibility: 1

Testing: 1

Documentation: 1

Format: 0