

# Project 4

Isaac Dudney

April 11, 2018

## 1 Psuedocode

A general description of how my code works is simple. Each of the cache replacement algorithms is represented by a Pure Function which takes two lists as input and returns a single list. The two list inputs are the pattern (eg ABCDEABC) and the available slots (which are initialized to None and replaced as the algorithm fills them). Each function initializes a special list of strings which store the text output, which is returned at the end of the function. Every function iterates over the pattern character by character and performs comparisons on the slots to determine behavior. It then appends output text to the list of strings according to the behavior; for example, if it gets a cache hit, it appends a + to the appropriate slot.

The main thread starts by taking user input; it exits if the input isn't integers, or if the input is out of bounds. It then initializes each necessary item (pattern and slots) using the input integers. The functions are then defined, along with a utility function to properly print and wrap the output.

Each algorithm is then called and printed using the utility function, after which the program exits.

However, I will also write psuedocode for each algorithm, including the printing utility function, here in a set of listings. Starting with **FIFO**.

```
1 def func(x):  
2     return(x)
```