

Homework 16: Stacks & Queues

Part 1. Implement a symbol balance checker function for the Pascal programming language. Pascal allows for the following pairs: {}, (), [], begin end . All programs will begin with the word "begin" and end with the word "end". Your function should receive an ifstream object which is already open and will return true, all of the symbols match, or false, they do not. You do not have to worry about comments in the program but you do have to avoid other parts of the program's code such as assignment statements (x=y) and other expressions. Example:

```
begin {} end returns True
```

```
begin [{ }()] end           returns False
```

Please submit a full program for this like with your previous assignments; with a main function to show that your function works.

Part 2. Although a queue is "best" implemented with a list, it can be implemented with a vector if you take into account the starting position of the queue. For example, if five elements are pushed onto the queue, the start of the queue is at position zero and the end is at position 4. If we, then, pop two elements, the start would be at position 2 and the end at position 4. The two "popped" elements are not really removed from the vector, and that avoids the $O(N)$ time problem for the pop function.

Implement a class which uses a vector to store the queue. Be mindful of performance, such that if the queue is empty, the size of the underlying vector is "reset." You do not need to submit a main function for this, just the class is fine.