Assignment 1: Design
Joshua Dong
January 25, 2015

1 System IPO

Input	Processing	Output
Input filename	Concurrently find palindromes line-by-line	List of palindromes joined by spaces, per line
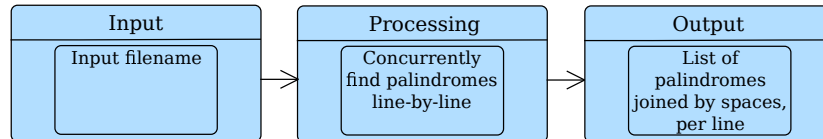
2 Functional Block Diagram

readFile(filename)

↓

parse(line)

↓ ↓

searchOddPalindromes(line)   searchEvenPalindromes(line)

↓ ↓

removeSubPalindromes(results)

↓

makeOutput(palindromes)

↓

More lines? — yes
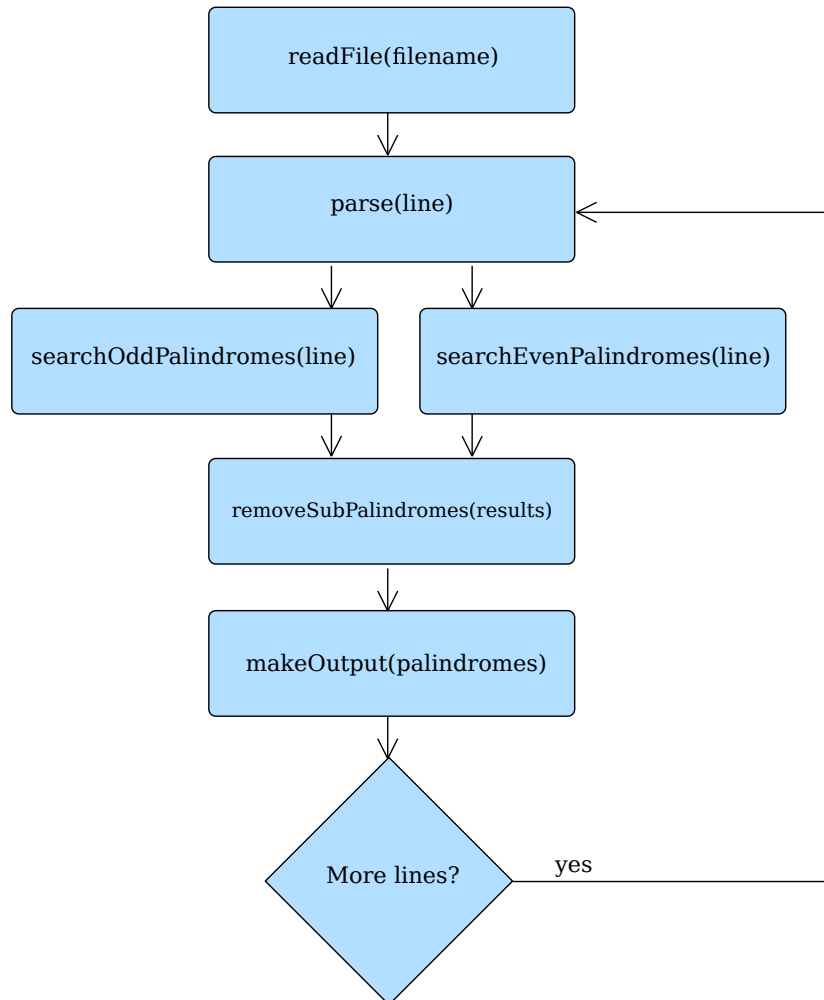
3 Pseudocode

```
readFile(Filename) ->
    Lines := open(Filename, 'r').read().split('\n'),
    Lines[:"QUIT"].

isPalindrome(String) ->
    case length(String) rem 2 of
        0 ->
        String[:length(String)/2] == reverse(String[length(String)/2:]);
        1 ->
        String[:length(String)/2] == reverse(String[(length(String)/2)+1:])
    end.

evenSearch(Line, Results) ->
    do stuff,
    ++Semaphore;

oddSearch(Line, Results) ->
    do stuff,
    ++Semaphore;

parseLine(Line) ->
    Semaphore = -1,
    Even = spawn(evenSearch(Line, Results, Semaphore)),
    Odd = spawn(oddSearch(Line, Results, Semaphore)),

    Semaphore.aquire();
    Results := filter(Results, (Palindrome) -> !isSubPalindrome(Palindrome)),
    Results.

main([Filename]) ->
    Lines := readFile(Filename),
    LineResults := [parseLine(Line) for Line in Lines],
    for LineResult in LineResults:
        print(''.join(LineResult))
```