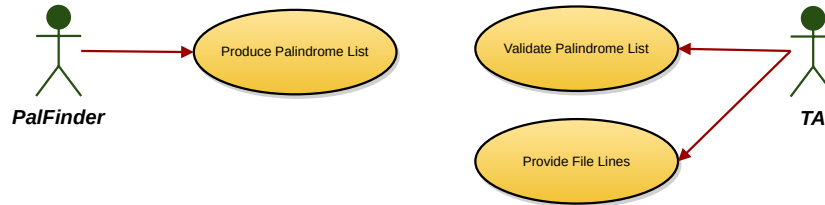


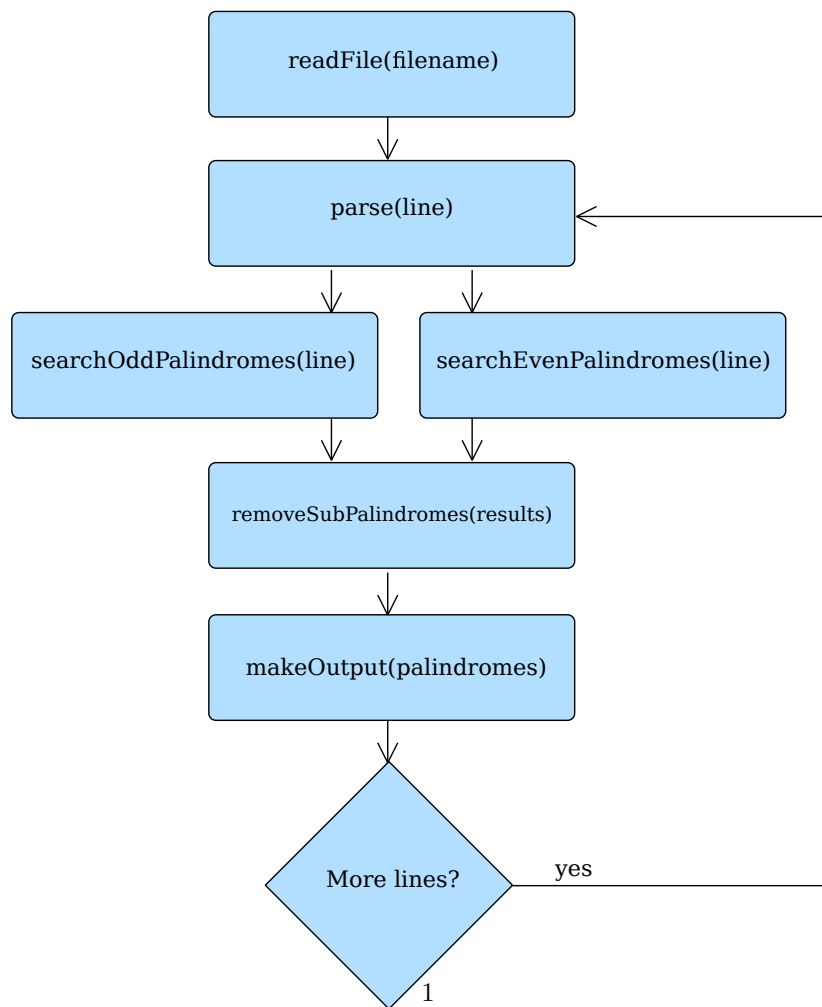
# Assignment 1: Design

Joshua Dong  
January 28, 2015

## 1 UML Diagram



## 2 Functional Block Diagram



### 3 Pseudocode

```
def is_palindrome(s):
    return s[:len(s)//2] == s[:len(s)//2 - (len(s) % 2 - 1):-1]

def even_search(s):
    results = set()
    for i in range(1, len(s)-2):
        max_search = min(i, len(s) - 2 - i)
        best_palindrome = None
        for l in range(max_search):
            substring = s[i - l - 1: i + l + 3]
            if is_palindrome(substring): best_palindrome = substring
            else: break
        if best_palindrome:
            results.add(best_palindrome)
    return results

def odd_search(s):
    results = set()
    for i in range(1, len(s)-1):
        max_search = min(i, len(s) - 1 - i)
        best_palindrome = None
        for l in range(max_search):
            substring = s[i - l - 1: i + l + 2]
            if is_palindrome(substring): best_palindrome = substring
            else: break
        if best_palindrome:
            results.add(best_palindrome)
    return results

def is_sub_palindrome(test, palindrome):
    if len(test) < len(palindrome) and len(test) % 2 == len(palindrome) % 2:
        offset = (len(palindrome) - len(test)) // 2
        if palindrome[offset:offset + len(test)]:
            return True
    return False

def is_sub_palindrome_pool(test, pool):
    for p in pool:
        if is_sub_palindrome(test, p): return True
    return False

def parse(s):
    results = list(even_search(s) | odd_search(s))
    print(' '.join(sorted([_ for _ in results\
                           if not is_sub_palindrome_pool(_, results)])))
```