论题 1-12 作业

姓名：陈劭源　　学号：161240004

1 [DH] Problem 5.4

(a)

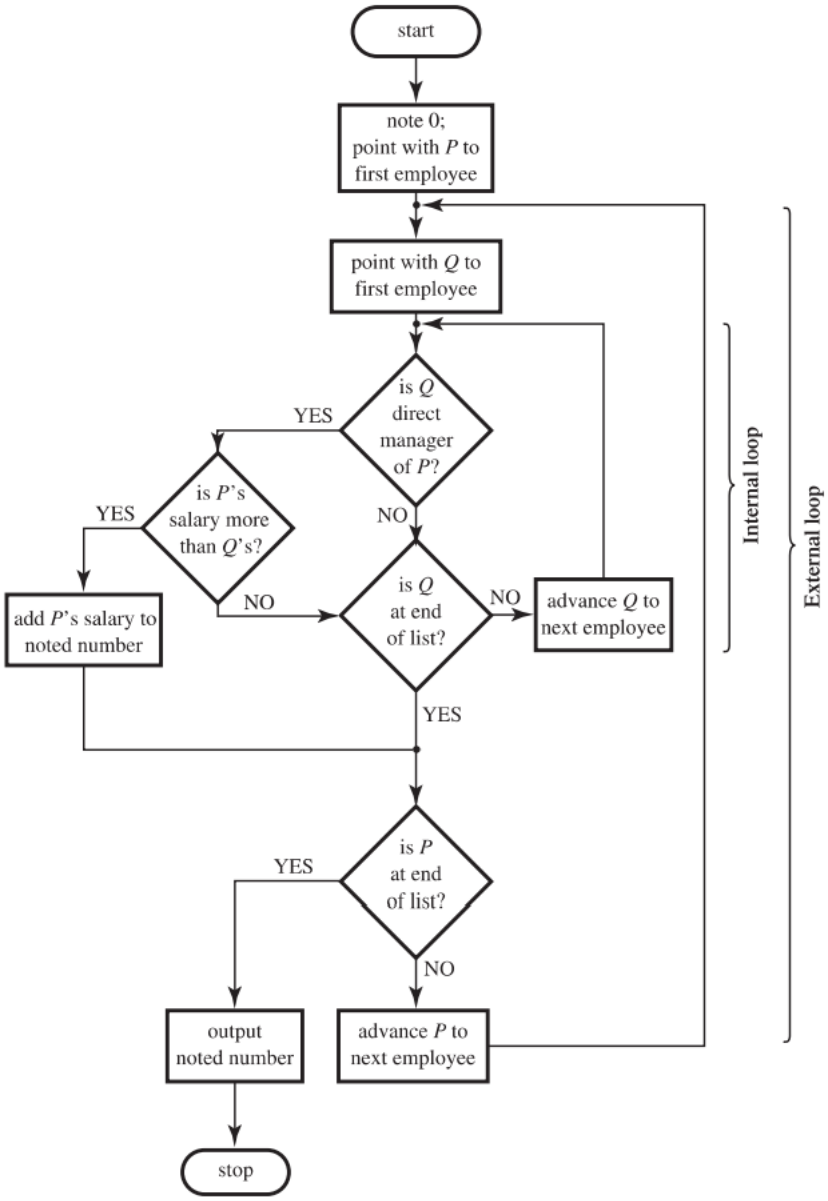

Figure 3: modified flowchart

(b) 部分正确性的证明：在外层循环前断言（记为 A）“当前的数字是前 M 位员工中，比至少一个直接上级工资高的那些员工的工资总和”，其中 M 是外层循环已经执行的次数；在内层循环前断言（记为 B）“前 N 位雇员要么不是 P 的直接上级，要么虽然是 P 的直接上级，但工资不比 P 低”。断言 A 在第一次循环执行前是正确的，因为当前数字被初始化为 0；断言 B 在第一次循环执行前是显然正确的。当内层循环执行一次后，如果没有退出循环，即已确认 Q 不是 P 的直接上级，或者 Q 虽然是 P 的直接上级，但工资不比 P 低，断言 B 依旧成立；如果退出了循环，则说明 Q 是 P 的直接上级且工资比 P 低，并且 P 的工资已加入当前数字之中，断言 A 依旧成立；如果因 Q 是最后一个员工而结束了循环，则说明所有员工都不是比 P 工资低的直接上级，并且当前数字无变化，断言 A 依旧成立。当外层循环执行结束后，断言 A 指出，输出的数字是所有员工中，比至少一个直接上级工资高的那些员工的工资总和，这是符合要求的正确输出。

可终止性的证明：对于内层循环，指针 Q 未指向过的员工数量每次循环之后会减少，且最少为 0，因此内层循环能够终止；对于外层循环，指针 P 未指向过的员工数量也有类似的性质，因此外层循环也能够终止。总之，对于任意合法的输入，该算法总是能够终止。

这一算法既是部分正确的又是可终止的，因而是完全正确的。

(c) Yes.

(d) No.

2 [DH] Problem 5.6

(a) Only three well-placed invariants are sufficient for the proof of the partial correctness of an algorithm which contains only one loop.

(b) The flowchart that doesn't contain any loop.

(c) 4.

(d) 4.

3 [DH] Problem 5.8

function **rev**(X)

$T \leftarrow X$;

$Y \leftarrow \Lambda$;

while $T \neq \Lambda$ do the following:

$Y \leftarrow Y \cdot$ **last**(T);

$T \leftarrow$ **all-but-last**(T);

return Y.

部分正确性的证明：在每次循环执行前断言“$X = T \cdot \mathbf{rev}(Y)$”。第一次循环执行之前，$T$ 被设置为 X，Y 被设置为空串，因而断言成立；循环每执行一次后，T 少了最后一个字符，而这一字符被加到 Y 的最后，因而断言仍然成立。循环结束后，T 为空串，并且断言依然成立，从而有 $Y = \mathbf{rev}(X)$，这是正确的输出。

可终止性的证明：每次执行循环后，T 的字符数会少 1，并且当字符数为 0 时循环终止；而对于合法的输入，T 的字符数在第一次循环执行之前是有限的，因此算法总是能够终止。

综上，该算法是完全正确的。

4 [DH] Problem 5.9

function **equal**(X, Y)

$P \leftarrow X$;

$Q \leftarrow Y$;

while $P \neq \Lambda$ and $Q \neq \Lambda$ do the following:

if **eq** (**head**(P), **head**(Q)) is false then return false;

$P \leftarrow$ **tail**(P);

$Q \leftarrow$ **tail**(Q);

if $P = \Lambda$ and $Q = \Lambda$ then return true;

otherwise return false.

部分正确性的证明：在每次循环执行前断言“X 和 Y 的前 N 个字符相同，并且 P, Q 分别是 X, Y 除去 N 个字符后的剩余部分”，其中 N 是循环已经执行的次数。当第一次循环开始执行前，断言显然是成立的。当循环执行了一次后，若循环没有退出，X 和 Y 的第 $N+1$ 个字符相同，并且 P, Q 的首字符被去掉了，因而断言依然成立；若循环退出，则 P 和 Q 的首字母不同，也就是说，X 和 Y 在某个位置处的字母不同，因此整个字符串也不同，算法给出了正确的输出。当循环结束时，P 和 Q 其中有一个是空串，若两个都是空串，则这两个字符串相等，算法给出了正确的结果；若有一个不是空串，根据断言可知，X 和 Y 的字符个数不相等，字符串本身当然不相等，算法的输出也是正确的。

可终止性的证明：每次循环执行后，P 和 Q 的字母个数少了 1，当 P 和 Q 的字母个数有一个为 0 时，整个算法终止。而对于合法的输入，字符串 P 和 Q 的字母个数都是有限的，因而算法是可终止的。

综上，该算法是完全正确的。

5 [DH] Problem 5.10

(a) 对于任意合法的输入 S，根据回文串的定义，当它自身和它的逆序串相等时，S 是回文串，否则不是，因此该算法是部分正确的。该算法不含循环结构，并且上面已经证明了，**rev** 和 **equal** 都是可终止的，因而该算法是可终止的。综上，该算法是完全正确的。

(b) 如果程序 A 中不含循环结构，程序 A 用合法的参数调用了程序 B，并且程序 B 是可终止的，那么程序 A 也是可终止的。

6 [DH] Problem 5.11

(a) `"abcdefghijklmnopqrstuvwxyz"`.

(b) 3（**eq**, **head**, **last** 各 1 次）

7 [DH] Problem 5.12

(a) 正确。定义 **left**(S, i) 为字符串 S 最开始的 i 个字符构成的字符串，**right**(S, i) 为字符串 S 最后的 i 个字符构成的字符串，**len**(S) 为字符串 S 中字符的个数。在每次执行循环前，断言“**left**(S, i)=**rev**(**right**(S, i))，其中 i 为语句‘$X \leftarrow$ **all-but-last**(**tail**(X))’执行的次数”。在第一次循环执行之前，断言显然是成立的。当循环执行了一次之后，如果该次循环执行时 **eq**(**head**(X), **last**(X)) 成立，那么对该次循环执行之前的 X （记为 X'），有 **left**(S·**head**(X'),i)=**rev**(**right**(S·**tail**(X'), i))，因此断言依然成立；反之，断言未发生变化，依然是成立的。当循环结束后，根据断言有 **left**(S, $\lceil$**len**(S)/2$\rceil$)=**rev**(**right**(S, $\lceil$**len**(S)/2$\rceil$))，从而有 S =**rev**(S)。在这种情况下，循环时 **eq**(**head**(X), **last**(X)) 总是成立的，因而输出总是 true，这是正确答案。

(b) 错误。对于输入 `"ab"`，每次执行循环时，**eq**(**head**(X), **last**(X)) 总不成立，因而 X 的长度不会发生变化，循环不会终止。

8 [DH] Problem 5.13

(a) 不正确。对于输入 `"a"`，第一次执行循环时，X =“a”，$Y = \Lambda$，**eq**(**head**(X), **last**(Y)) 不成立，循环结束，输出 false. 但是 `"a"` 是一个回文串。

(b) 正确。每次执行循环后，如果 **eq**(**head**(X), **last**(Y)) 成立，X 的长度减少，循环继续执行；否则退出推出循环。对于合法的输入，字符串 X 的长度总是有限的，因而算法总是能够结束。

9 [DH] Problem 5.14

(a) $X \leftarrow S$;
$E \leftarrow$ true;
while $X \neq \Lambda$ and E is true do the following:
if **eq**(**head**(X), **last**(X)) then $X \leftarrow$ **all-but-last**(**tail**(X));
otherwise $E \leftarrow$ false;

return E.

(b) 部分正确性的证明：在每次执行循环前，断言“如果 E 为 true，那么 **left**(S, i)=**rev**(**right**(S, i))，其中 i 为语句‘$X \leftarrow$ **all-but-last**(**tail**(X))’执行的次数；否则 S 不是回文串”。在第一次循环执行之前，可以验证断言成立；在循环被执行了一次之后，如果该次循环执行时 **eq**(**head**(X), **last**(X)) 成立，那么对该次循环执行之前的 X （记为 X'），有 **left**(S·**head**(X'),i)=**rev**(**right**(S·**tail**(X'), i))，因此断言依然成立；否则，即可判定 S 不是回文串，断言也成立，并且下一次循环不会被执行。当循环结束后，根据断言，当 E 为 true 时有 **left**(S, $\lceil$**len**(S)/2$\rceil$)=**rev**(**right**(S, $\lceil$**len**(S)/2$\rceil$))，从而有 S =**rev**(S)，即 S 是回文串；当 E 为 false 时 S 不是回文串，因此算法能够给出正确的输出。

可终止性的证明：每次执行循环后，如果 **eq**(**head**(X), **last**(Y)) 成立，X 的长度减少，循环继续执行；否则退出循环。对于合法的输入，字符串 X 的长度总是有限的，因而算法是可终止的。

综上，该算法是完全正确的。

(c) 使用 *Pal*1 算法判断时，需要将整个字符串反转后再和原字符串比较，消耗的时间和字符串的长度成正比；而使用 *Pal*4 算法判断时，只需判断首尾两个字符即可断定字符串不是回文串，而无需继续比较，因此效率较高。