Homework 1

Question 2.2

The well-formed lists are:

((A) (B))
(A (B (C)))
(((A) (B)) (C))

Question 2.4

The parenthesis Notation is: (BOWS ARROWS) (FLOWERS CHOCOLATES)

Question 2.6

() → NIL
(()) → (NIL)
((())) → ((NIL))
(() ()) → (NIL NIL)
(() (())) → (NIL (NIL))

Question 2.13

(((FUN)) (IN THE) (SUN)):

1. **FUN**:
    - Start: (((FUN)) (IN THE) (SUN))
    - CAR: ((FUN))
    - CAAR: (FUN)
    - CAAAR: FUN
2. **IN**:
    - CDR: ((IN THE) (SUN))

- CAR: (IN THE)
- CAAR: IN

3. **THE**:
    - CDR: ((IN THE) (SUN))
    - CAR: (IN THE)
    - CADR: THE
4. **SUN**:
    - CDDR: ((SUN))
    - CAR: (SUN)
    - CAAR: SUN

Question 2.15

CAR	(A B)
CDDR	((E F))
CADR	(C D)
CDAR	B
CAADR	C
CDDAR	F
CAAR	A
CDADDR	NIL
CADDDR	F

Question 2.16

When given the input (FRED NIL) CAAR does FRED because:

CAR of (FRED NIL) is FRED

CAR of FRED is FRED itself

Sebesta Chapter 1:

6. Most UNIX is written in C

7. The disadvantage of having too many features in a language is that it can make it complex and difficult to learn. It also reduces the readability of it.

8. User-defined operator overloading can harm readability by making it unclear what an operator does, especially if the behavior is in non-standard ways.

9. In C, arrays and pointers are closely related, but their use is not fully orthogonal

10. The language ALGOL 68 used orthogonality as a primary design criterion.

11. The GOTO statement is used as a primitive control statement to build more complicated control structures.

12. A program is reliable if it performs its intended functions correctly and consistently under specified conditions

13. It's important because it prevents type errors otherwise it can lead to unexpected behavior or crashes

14. Aliasing occurs when two or more different names refer to the same memory location

15. Exception handling is a feature in programming languages that allows programs to detect and respond to runtime errors

16. Readability is important to writability because clear and easily understandable code is easier to write, modify, and debug.

20. The two are: poor support for modularity and lack of data abstraction

21. encapsulation, Inheritance, and Polymorphism.

22. Language Simula

23. Efficiency and readability.

24. Compilation, Interpretation, and Hybrid implementation.

25. A compiler

29. Easier debugging, platform independence, immediate execution which does not require a separate compilation step