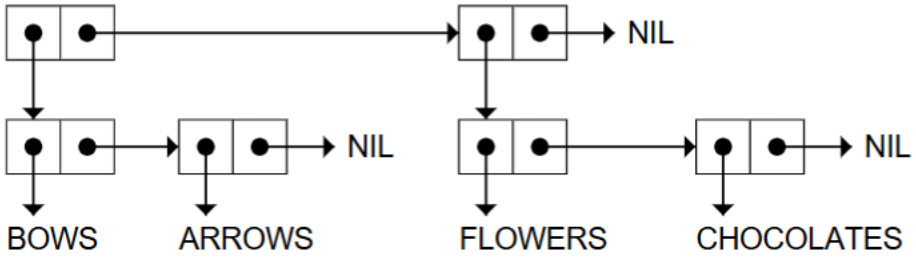
Question#1

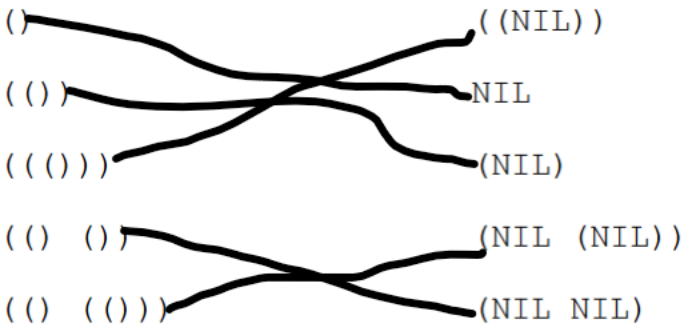
2.2. Which of these are well-formed lists? That is, which ones have properly balanced parentheses?
(A B (C)
((A) (B))--------well-formed
A B)(C D)
(A (B (C))
(A (B (C)))------- well-formed
(((A) (B)) (C))------ well-formed

2.4. What is the parenthesis notation for this cons cell structure?



- ((BOWS ARROWS) (FLOWERS CHOCOLATES))

2.6. Match each list on the left with a corresponding list on the right by substituting NIL for ( ) wherever possible. Pay careful attention to levels of parenthesization.



2.13. Write down tables similar to the one above to illustrate how to get to each word in the list (((FUN)) (IN THE) (SUN)).

1. Extracting FUN

- C..AR of the list → ((FUN))

- C.AAR of that result → (FUN)
- CAAAR of that result → FUN

2. Extracting IN

- C..DR of the list → ((IN THE) (SUN))
- C.ADR of that result → (IN THE)
- CAADR of that result → IN

3. Extracting THE

- C…DR of the list → ((IN THE) (SUN))
- C..ADR of that result → (IN THE)
- C.DADR of that result → (THE)
- CADADR of that result → THE

4. Extracting SUN

- C…DR of the list → ((IN THE) (SUN))
- C..DDR of that result → ((SUN))
- C.ADDR of that result → (SUN)
- CAADDR of that result → SUN

2.15. Using the list ((A B) (C D) (E F)), fill in the missing parts of this table.

Function	Result
CAR	(A B)
CDDR	((E F))
CADR	(C D)
CDAR	(B)
CADAR	B
CDDAR	NIL
CAAR	A
CDADDR	(F)
CADADDR	F

2.16. What does CAAR do when given the input (FRED NIL)?

CAAR means **"CAR of the CAR"**. First, apply CAR to (FRED NIL), which gives FRED.

FRED is just a symbol, applying CAR to it results in an **error**.

Question#2

6. In what language is most of UNIX written?
Most of UNIX is written in **C**.

7. What is the disadvantage of having too many features in a language?
Having too many features can make the language:

1. Complex and hard to learn for beginners.
2. Difficult to maintain due to increased potential for overlapping functionality and redundancy.
3. Less readable, as different developers may use different features to achieve the same goal.

8. How can user-defined operator overloading harm the readability of a program?
User-defined operator overloading can harm readability if:

1. Operators are given unexpected meanings, making code behavior unpredictable.
2. Overloading is done inconsistently or excessively, leading to confusion about the operator's purpose in different contexts.

9. What is one example of a lack of orthogonality in the design of C?
In C, **arrays and pointers are not completely orthogonal**:

1. Arrays are automatically passed by reference, while other data types are not.
2. The syntax for array indexing and pointer arithmetic overlaps, but with subtle differences.

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

11. What primitive control statement is used to build more complicated control statements in languages that lack them?

The GOTO statement is a primitive control statement used to build more complicated control structures like loops or conditional branches in some languages.

12. What does it mean for a program to be reliable?
A program is reliable if:

1. It performs its intended functions under predefined conditions.
2. It handles errors gracefully, avoiding crashes or unexpected behavior.

13. Why is type checking the parameters of a subprogram important?
Type checking ensures that:

1. Correct types of arguments are passed, reducing errors at runtime.
2. Programs are more robust and maintainable by catching mismatches during compilation.

14. What is aliasing?
Aliasing occurs when two or more variables refer to the same memory location. This can lead to unintended side effects, making debugging more difficult.

15. What is exception handling?
Exception handling is a mechanism for managing runtime errors or other exceptional conditions in a controlled way, ensuring the program continues to run or exits gracefully.

16. Why is readability important to writability?
Readability improves writability because:

1. Clear and well-documented code is easier to understand and extend.
2. Developers can build on existing readable code efficiently, avoiding errors.

20. What two programming language deficiencies were discovered as a result of the research in software development in the 1970s?

1. Inadequate support for data abstraction: Languages lacked mechanisms for defining and using abstract data types effectively.
2. Poor control over concurrency: Existing languages did not provide robust support for managing concurrent processes.

21. What are the three fundamental features of an object-oriented programming language?

1. Encapsulation**:** Combining data and the methods that operate on it into a single unit (class).
2. Inheritance**:** Allowing new classes to inherit properties and methods from existing ones.
3. Polymorphism**:** Enabling a single function or method to operate on different types of objects.

22. What language was the first to support the three fundamental features of object-oriented programming?

The first language to support the fundamental features of object-oriented programming was Simula 67.

23. What is an example of two language design criteria that are in direct conflict with each other?

1. Reliability vs. Efficiency: Ensuring a program runs reliably often involves additional checks, error handling, and safety features, which can reduce runtime efficiency.
2. Readability vs. Writability: A language optimized for ease of reading might use verbose syntax, making it harder to write concisely, and vice versa.

24. What are the three general methods of implementing a programming language?

1. Compilation: Translating the entire source code into machine code before execution.
2. Interpretation: Executing the source code line by line without converting it into machine code.
3. Hybrid Implementation: Combining compilation and interpretation, where the code is partially compiled into an intermediate form, which is then interpreted.

25. Which produces faster program execution, a compiler or a pure interpreter?

A compiler produces faster program execution because it translates the code into machine language before execution, whereas a pure interpreter executes code line by line, incurring runtime overhead.

29. What are the advantages in implementing a language with a pure interpreter?

1. Immediate Execution and Testing:
   Programs can be executed immediately without the need for a separate compilation step, making the development process faster, especially for debugging.
2. Platform Independence:
   Since the interpreter executes code directly, it avoids dependency on specific hardware or compiled binaries, making the program portable.
3. Dynamic Features:
   Pure interpreters can handle dynamic features like runtime code modifications or dynamic typing more effectively than compilers.
4. Error Detection and Debugging:
   Errors are detected at the time of execution, providing real-time feedback to developers about issues in the code.