**Kevin Gutierrez – Homework 1 – CSCI 330**

**Question 1:**

2.2:

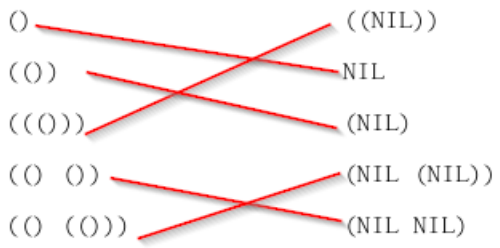
1. ((A) (B))
2. (A (B (C)))
3. (((A) (B)) (c))

2.4: ((BOWS ARROWS) (FLOWERS CHOCOLATES))

2.6:

**EXERCISE**

**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:

| ***FUN*** **Step** | **Result** |
|---|---|
| *Start* | (((FUN))(IN THE)(SUN)) |
| C...DR | ((FUN)) |
| C..AAR | (FUN) |
| CAAAR | FUN |

| ***IN*** **Step** | **Result** |
|---|---|
| *Start* | (((FUN))(IN THE)(SUN)) |
| C...DR | ((IN THE)(SUN)) |
| C..ADR | (IN THE) |
| CAADR | IN |

| ***THE*** **Step** | **Result** |
|---|---|
| *Start* | (((FUN))(IN THE)(SUN)) |
| C...DR | ((IN THE)(SUN)) |
| C..ADR | (IN THE) |

| | |
|---|---|
| CADADR | THE |

| ***SUN*** **Step** | **Result** |
|---|---|
| *Start* | (((FUN))(IN THE)(SUN)) |
| C...DR | ((IN THE)(SUN)) |
| C..DDR | ((SUN)) |
| CADDR | (SUN) |
| CAADDR | SUN |

2.15:

**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: It will error out because a CAR operation cannot be done on a single element (CAR only works on lists)

**Question 2:**

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

**7. What is the disadvantage of having too many features in a language** – It makes the language harder to learn

**8. How can user-defined operator overloading harm the readability of a program?** – It can reduce readability when users don't do it sensibly, which could confuse users of the overloaded operator.

**9. What is one example of a lack of orthogonality in the design of C?** – In C, structs can be returned from functions but arrays cannot.

**10. What language used orthogonality as a primary design criterion?** – ALGOL

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

**12. What does it mean for a program to be reliable** – A program is reliable if it performs to its specifications under all conditions

**13. Why is type checking the parameters of a subprogram important?** – It's important because it can lead to runtime errors, leading to undetectable inconsistency by the compiler or runtime system.

**14. What is aliasing** – Having two or more distinct names in a program that can be used to access the same memory cell, such as two unique pointers pointing to the same variable.

**15. What id exception handling?** – Exception handling is the ability of a program to intercept runtime errors (as well as other unusual conditions detectable by the program), take corrective measures, and continue.

**16. Why is readability important to writability** – Programs that are difficult to read are both difficult to write and modify

**20. What two programming language deficiencies were discovered as a**

**result of the research in software development in the 1970s?** – Incompleteness of type checking and inadequacy of control statements

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

Data abstraction (encapsulation), inheritance, and dynamic method binding (polymorphism)

**22. What language was the first to support the three fundamental features of**

**object-oriented programming?** - SmallTalk

**23. What is an example of two language design criteria that are in direct**

**conflict with each other?** – Reliability and cost of execution

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

**language?** – Compiler implementation, pure interpretation, hybrid interpretation

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

**interpreter?** - Compiler

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

**interpreter?** – Source level debugging operations, ease of implementation, and flexibility for dynamic features

**Question 3:**

Sequence of CARs and CDRs that return x when applied to:

a. (a b x d)
b. (a (b (x d)))
c. (((a (b (x) d))))

**for a:** (car (cdr (cdr '(a b x d ))))

**for b:** (car (car (cdr (car (cdr '(a (b (x d ))))))))

**for c:** (car (car (cdr (car (cdr (car (car '(((a (b (x) d ))))))))))))

**Question 4:**

**for a:** (cons 'a (cons 'b (cons 'x (cons 'd nil))))

**for b:** (cons 'a (cons (cons 'b (cons (cons 'x (cons 'd nil )) nil )) nil ))

**for c:** (cons (cons (cons 'a (cons (cons 'b (cons (cons 'x nil) (cons 'd nil))) nil)) nil) nil)