<Homework 1> Solve the Problems Chapter 2 - #7, #12, #14

#7. LISP began as a pure functional language but gradually acquired more and more imperative features. Why?

 실제로 존재하는 대부분의 알고리즘들이 imperative 형식으로 작성되었다는 점, 코드를 작성할 때 imperative방식을 바탕으로 효과적인 코드를 작성할 수 있다는 점에서 LISP에 발전함에 따라 imperative features가 추가되었다.

#12. Describe, in your own words, the concept of orthogonality in programming language design.

기본적으로 Orthogonal은 수학에서 두 벡터, 혹은 공간 사이에 직교함을 의미한다. 프로그래밍에서 orthogonality는 두 가지 의미로 존재한다. 첫 번째로, 컴퓨터 프로그래밍에서 orthogonality란, 한 수행이 다른 작업들에 영향을 주지 않는 것을 의미한다. 이와 다르게 프로그래밍 언어에서 orthogonality란, 한 프로그래밍 언어에 대해, small set of primitive constructs(적은 수의 기능들)을 통해 서로 조합을 했을 때 예외가 존재하지 않는 언어를 의미한다. 예를 들어, C 언어에서 for, while은 둘 다 반복문이라는 점에서 overlapping하며, if와 for은 서로 orthogonal 하다. 하지만, C 언어는 return() 기능과 data type 중 array가 예외가 되기 때문에 orthogonal하지 않다고 볼 수 있다. 이와 다르게, python은 return()에 int, float 뿐만 아니라, list도 적용할 수 있기 때문에, 위 경우에 대해 orthogonal하다고 볼 수 있다.

#14. What are the arguments both for and against the idea of a typeless language?

typeless language의 한 예시로는, assembly language가 있다. 이러한 typeless language는 당연하게 type check를 수행하지 않기 때문에, static, 혹은 dynamic type binding으로 이루어지는 프로그래밍 언어보다 속도 면에서 빠를 수 있다. 또한 type이 존재하지 않기 때문에 다양한 subroutines 을 적용할 수 있는 flexibility 또한 존재한다. 하지만 단점으로는, 프로그램을 작성하는 사람에게 더 많은 책임을 부여한다는 점이 존재하며, 이는 프로그래밍 언어에 대해 reliability가 낮다는 말과 동일하다.