עבודת בית 1

מגישים: עומרי גולדברג 208938985, נעה חרדון 312164668

חלק ראשון :

1. Imperative- Control flow is an explicit sequence of commands- in contrast to "declarative"
   Procedural — Imperative programming organized around hierarchies of nested procedure calls.
   Functional — Computation proceeds by (nested) function calls that avoid any global state mutation and through the definition of function composition.

   Procedural paradigm improve the imperative paradigm- In imperative paradigm the program is build by commends. Performing complex operations requires an inefficient form of work (copy and paste code in order to do the same command on different parameters, Code repetition, Cannot be easily adapted to new values of the parameters, The nature of the task is not reflected in the structure of the code)
   The Procedural paradigm improve these issues by adding **Procedures** (commands with a well defined interface of input parameters / output parameters and expected behavior) and **Local variables** (variables which are defined only within the scope of the procedure)

   Functional paradigm improves the procedural paradigm- functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids state and mutable data (no side effects). It emphasizes the application of functions, in contrast with the procedural programming style that emphasizes changes in state.

2. (a) <T1, T2>(x: T1[], y: (pred: T2)⇒ Boolean)⇒ Boolean
   (b) x:number[]⇒number (x: number[ ] ⇒x.reduce((acc: number, cur: number) => number)
   (c) <T>(x:Boolean, y:T[])⇒T

   Abstraction barriers- isolate different ``levels'' of the system. At each level, the barrier separates the programs (above) that use the data abstraction from the programs (below) that implement the data abstraction- can be seen in collection of procedures only higher-level procedures can call lower-level procedures. The separation is between how the program was implemented and between the use of the client – which won't need to worry about the implementation details.