Task 1: Install Haskell - https://www.haskell.org/ghcup/install/

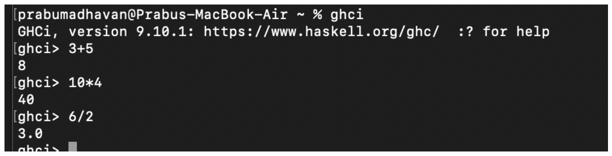
Lab 1: Introduction to Programming Paradigms (Simple Programs in Haskell)

Part 1: Haskell Exercises (45 minutes)

1. **Basic Arithmetic:**
   - **Objective**: Get familiar with GHCi and basic arithmetic operations.
   - **Exercise 1:** Open GHCi and perform basic arithmetic operations:
   -

     Haskell Code

     ```
     3 + 5
     10 * 4
     6 / 2
     ```

     Sample Output 

     ```
     prabumadhavan@Prabus-MacBook-Air ~ % ghci
     GHCi, version 9.10.1: https://www.haskell.org/ghc/  :? for help
     ghci> 3+5
     8
     ghci> 10*4
     40
     ghci> 6/2
     3.0
     ghci>
     ```

   - **Exercise 2:** Define a function to calculate the square of a number:
     - **Open Terminal.**
     - **Create a file with the .hs extension using nano (or your preferred text editor).**
     - **Write the following code** inside the square.hs file
     - Press Ctrl + X to exit nano.
     - Press Y to confirm saving the file, then press Enter to confirm the filename (square.hs).
     - Compile the program by running the following command:

       ```
       ghc -o square square.hs

       ./square
       ```

     Haskell Code

     ```
     square :: Int -> Int
     square x = x * x
     main :: IO ()
     main = print (square 5)
     ```

     - Test it with a few numbers: `square 5`, `square 10`, etc.

2. **Defining and Using Lists:**
   - **Objective**: Understand basic data structures like lists in Haskell.
   - **Exercise 3:** Create a list of numbers and compute the sum of the list:

     Haskell Code
     ```
     sumList :: [Int] -> Int
     sumList [] = 0
     sumList (x:xs) = x + sumList xs
     ```

     - Test with: `sumList [1, 2, 3, 4, 5]`

3. **Pattern Matching with Lists:**
   - **Objective**: Learn how pattern matching works in Haskell.
   - **Exercise 4:** Write a function to check if a list is empty:

     Haskell Code
     ```
     isEmpty :: [a] -> Bool
     isEmpty [] = True
     isEmpty _  = False
     ```

     - Test with: `isEmpty [1, 2, 3]` and `isEmpty []`.

4. **Simple IO Operations:**
   - **Objective**: Understand basic input and output in Haskell.
   - **Exercise 5**: Write a program that asks the user for their name and prints a greeting:

     ```
     haskell
     Copy code
     main :: IO ()
     main = do
         putStrLn "What is your name?"
         name <- getLine
         putStrLn ("Hello, " ++ name)
     ```

     - Run the program and interact with it.