**PCPF Lab**

**Lab Assignment number 05**

**Name:** Aamir Ansari **Batch:** A **Roll no.** 01

**Aim**: Write Haskell program for the following

**Problem Statement:**

1. Find whether the read number is even or odd. Display "Even" is the no read in is even or else "Odd".

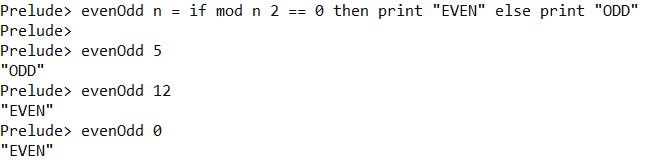
2. Generate Fibonacci series until a given number using recursive.

3. Find the sum of list of odd numbers in a list. Initialize the list. Report error if list is empty.

**Solution:**

Find whether the read number is even or odd. Display "Even" is the no read in is even or else "Odd"

evenOdd n = if mod n 2 == 0 then print "EVEN" else print "ODD"



Generate Fibonacci series until a given number using recursive.

fibonacci a b = a : fibonacci b (a+b)

main = do

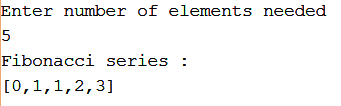
putStrLn "Enter number of elements needed"

input<-getLine

let n = (read input :: Int)

putStrLn "Fibonacci series : "

print (take(n) (fibonacci 0 1))



Find the sum of list of odd numbers in a list. Initialize the list. Report error if list is empty

sumOdd n | listSum n == 0 = print "EMPTY LIST"

| otherwise = print (listSum n)

listSum :: [Int]->Int

listSum n | n == [] = 0

| odd(head n) = (head n) + listSum(tail n)

| otherwise = 0 + listSum(tail n)