CS 3210 - Principles of Programming Languages (Spring 2020)

Activity 12

Haskell Practice on Math Functions

Implement the following functions in Haskell.

a) write function `isPrime` that takes an integer and returns true/false whether the input is a prime number or not.

b) write function `gcd` that takes two integers and returns the GCD (Greatest Common Divisor) of the two inputs.

c) write function `coprime` that takes two integers and returns true/false whether the numbers are coprimes (their GCD equals to 1).

d) write function `totientPhi` that takes an integer `m` and returns the number of positive integers `r` (`1 <= r < m`) that are coprime to m.

e) write function `primeFactors` that takes an integer and returns a flat list with the prime factors of the given number in ascending order.

f) write function `primeFactorsMult` similar to `primeFactors` but with the prime factors and their multiplicity.

g) write function `primesRange` that takes a range of integers and returns a list of all prime numbers within that range.

h) Goldbach's conjecture says that every positive even number greater than 2 is the sum of two prime numbers. Example: 28 = 5 + 23. It is one of the most famous facts in number theory that has not been proved to be correct in the general case. It has been numerically confirmed up to very large numbers. Write function `goldbach` that takes an integer and returns the two prime numbers that sum up to it.

i) write the function `golbachList` that takes a range of integers and returns a list of all even numbers and their Goldbach composition.