
CS 421 --- Introducing HASKELL via Project Euler

Manager	Keeps team on track	
Recorder	Records decisions	
Reporter	Reports to class	
Reflector	Assesses team performance	

Objectives

This guide will give you a quick introduction to HASKELL by walking through the solutions to a few Project Euler problems. The goal is not just to explain the language, but the thinking behind it.

Euler Problem 1 --- Multiples of 3 and 5

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000.

Here is the solution in HASKELL. This is presented in the Read Eval Print Loop, which you can run using `ghci` or `stack repl`. The string `Prelude>` is the prompt. Review this session with your team and answer the following questions.

```
1 Prelude> mod 10 3
2 1
3 Prelude> isValid x = x `mod` 3 * x `mod` 5 == 0
4 Prelude> isValid 3
5 True
6 Prelude> isValid 4
7 False
8 Prelude> [0..20]
9 [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
10 Prelude> filter isValid [0..20]
11 [0,3,5,6,9,10,12,15,18,20]
12 Prelude> euler1 = sum (filter isValid [0..999])
13 Prelude> euler1
14 233168
```

Problem 1) Line 1 and 3 both use `mod`, but line 3 uses backticks. What is the purpose of the backticks?

Problem 2) How would you describe the syntax of function calls in this language? Do you think it's okay to write `isValid(3)` instead of `isValid 3`?

Problem 3) The value `euler1` is hard-coded to go to 999. Modify it so that `euler1` accepts a parameter for the size of the range.

Euler Problem 3 --- Prime Factors

The prime factors of 13195 are 5, 7, 13 and 29. What is the largest prime factor of the number 600851475143?

Here is the code for filter, which you saw above.

```
1 filter f [] = []
2 filter f (x:xs) = if f x then x : rest
3                   else rest
4   where rest = filter f xs
```

Problem 4) What does the syntax `[]` mean? What does the syntax `x:y` mean? Hint: they mean different things depending on the context!

Problem 5) Consider the following code. How many arguments does the function `notDivides` take? What is happening in line 6?

```
1 Prelude> notDivides n x = x `mod` n > 0
2 Prelude> notDivides 4 10
3 True
4 Prelude> notDivides 5 10
5 False
6 Prelude> filter (notDivides 2) [2..10]
7 [3,5,7,9]
```

Problem 6) Here is a function that will perform the Sieve on a list. Only part of it is missing! Fill in the missing part.

```
1 emitAndRemove (n:xs) = n : emitAndRemove ( ... your code here ... )
```

Now let's see if it works!

```
1 Prelude> primes = emitAndRemove [2..]
2 Prelude> take 20 primes
3 [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71]
```

Problem 7) Describe the `take` function.

Problem 8) What is going on with `[2..]`?

Problem 9) What do you think would happen if we just asked for primes without using `take`?