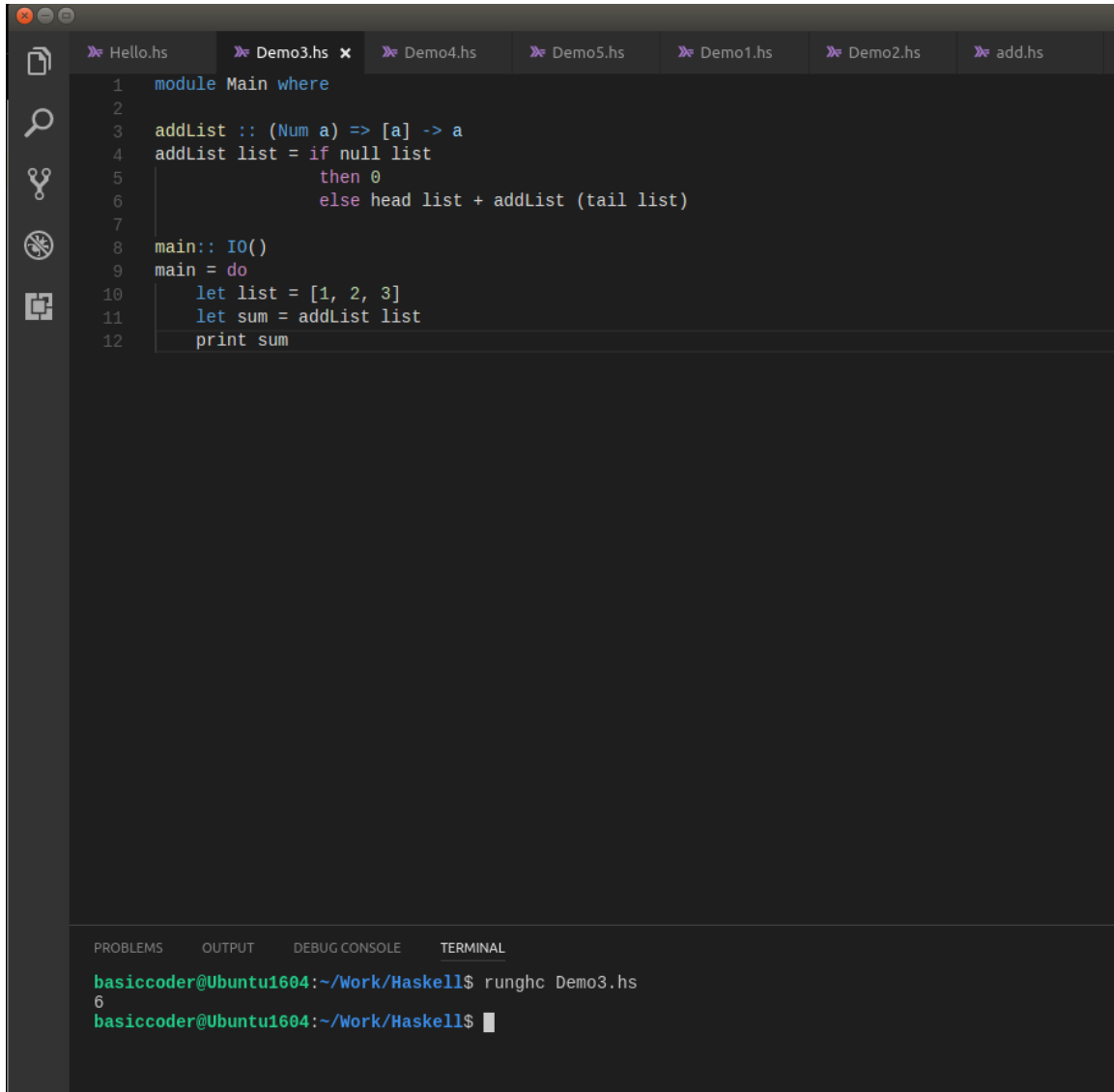


## 作业 1：用尾递归来实现列表求和

### 一：实验代码：



The image shows a screenshot of a Haskell IDE with a dark theme. The editor window displays a Haskell program in `Demo3.hs`. The code defines a function `addList` that calculates the sum of a list of numbers using tail recursion. The `main` function initializes a list `[1, 2, 3]`, calls `addList`, and prints the result. The terminal at the bottom shows the command `runghc Demo3.hs` being executed, resulting in the output `6`.

```
1 module Main where
2
3 addList :: (Num a) => [a] -> a
4 addList list = if null list
5               then 0
6               else head list + addList (tail list)
7
8 main :: IO()
9 main = do
10     let list = [1, 2, 3]
11     let sum = addList list
12     print sum
```

basiccoder@Ubuntu1604:~/Work/Haskell\$ runghc Demo3.hs  
6  
basiccoder@Ubuntu1604:~/Work/Haskell\$

## 二：实验结果：

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
basiccoder@Ubuntu1604:~/Work/Haskell$ runghc Demo3.hs
6
basiccoder@Ubuntu1604:~/Work/Haskell$
```

## 三：实现(x+y) by 1+ 函数(add1)

```
Hello.hs  Demo3.hs  Demo6.hs x  Demo4.hs  Dem
1  module Main where
2
3  add :: (Integral a) => a -> a -> a
4  add x y = add' x y 0
5
6  add' :: (Integral a) => a -> a -> a -> a
7  add' x y n = if n == y
8               then x
9               else add' (x+1) y (n+1)
10
11 main::IO()
12 main = do
13     print [(add 2 400)]

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
basiccoder@Ubuntu1604:~/Work/Haskell$ runghc Demo6.hs
402
basiccoder@Ubuntu1604:~/Work/Haskell$
```

附件：

[1] Demo3.hs

[2] Demo6.hs