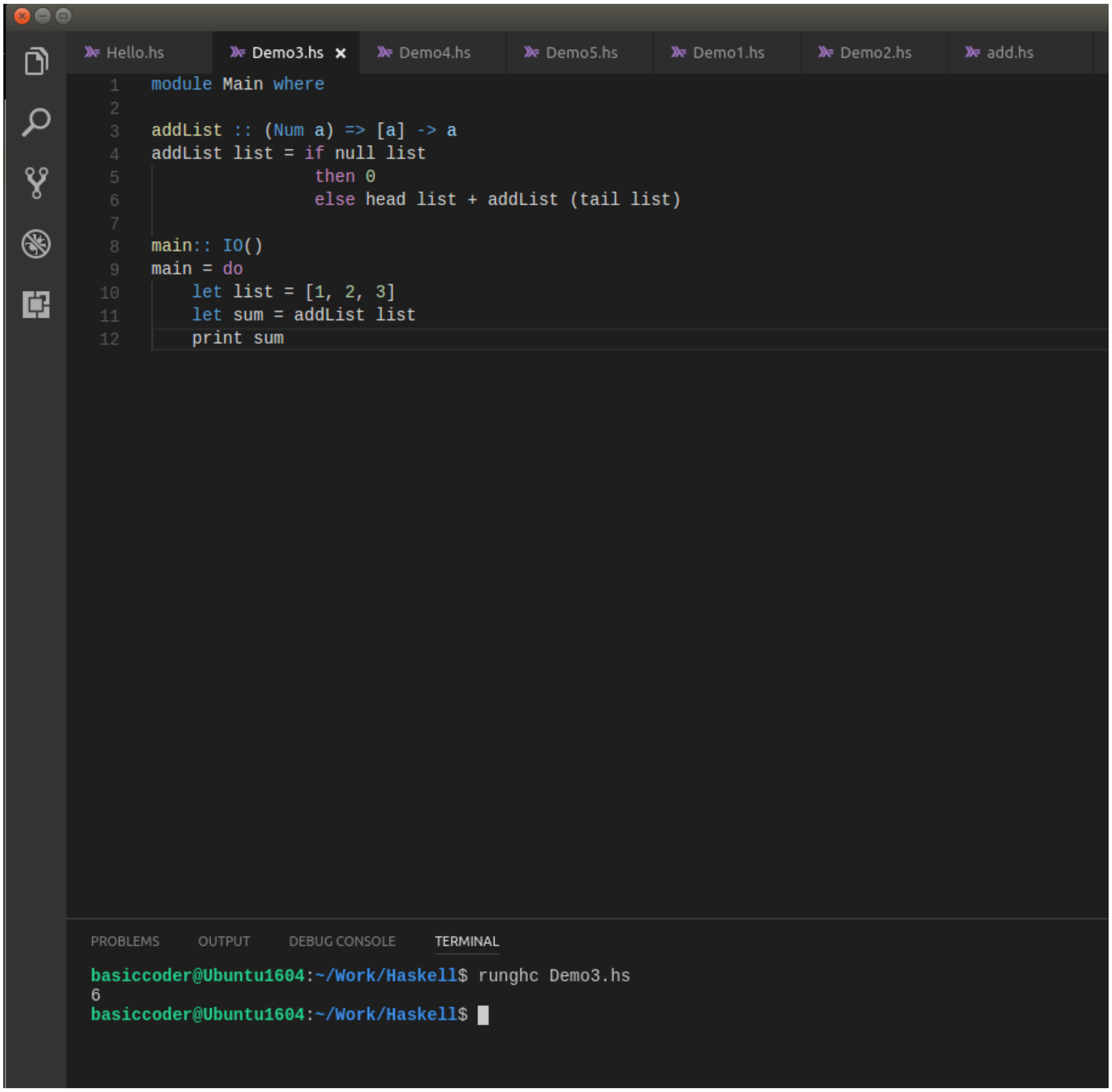
作业 1：用尾递归来实现列表求和

一：实验代码：



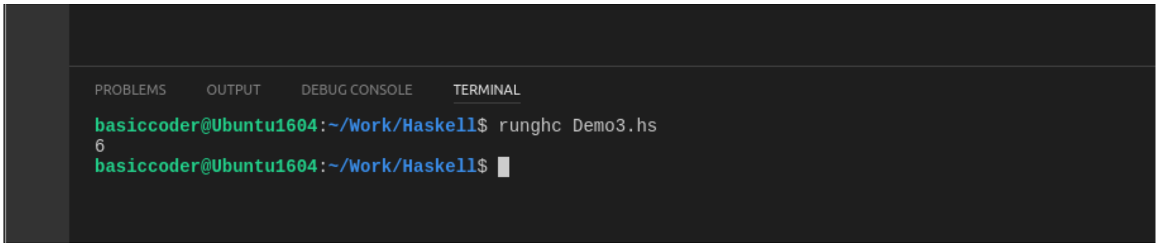
```haskell
module Main where

addList :: (Num a) => [a] -> a
addList list = if null list
                then 0
                else head list + addList (tail list)

main:: IO()
main = do
    let list = [1, 2, 3]
    let sum = addList list
    print sum
```

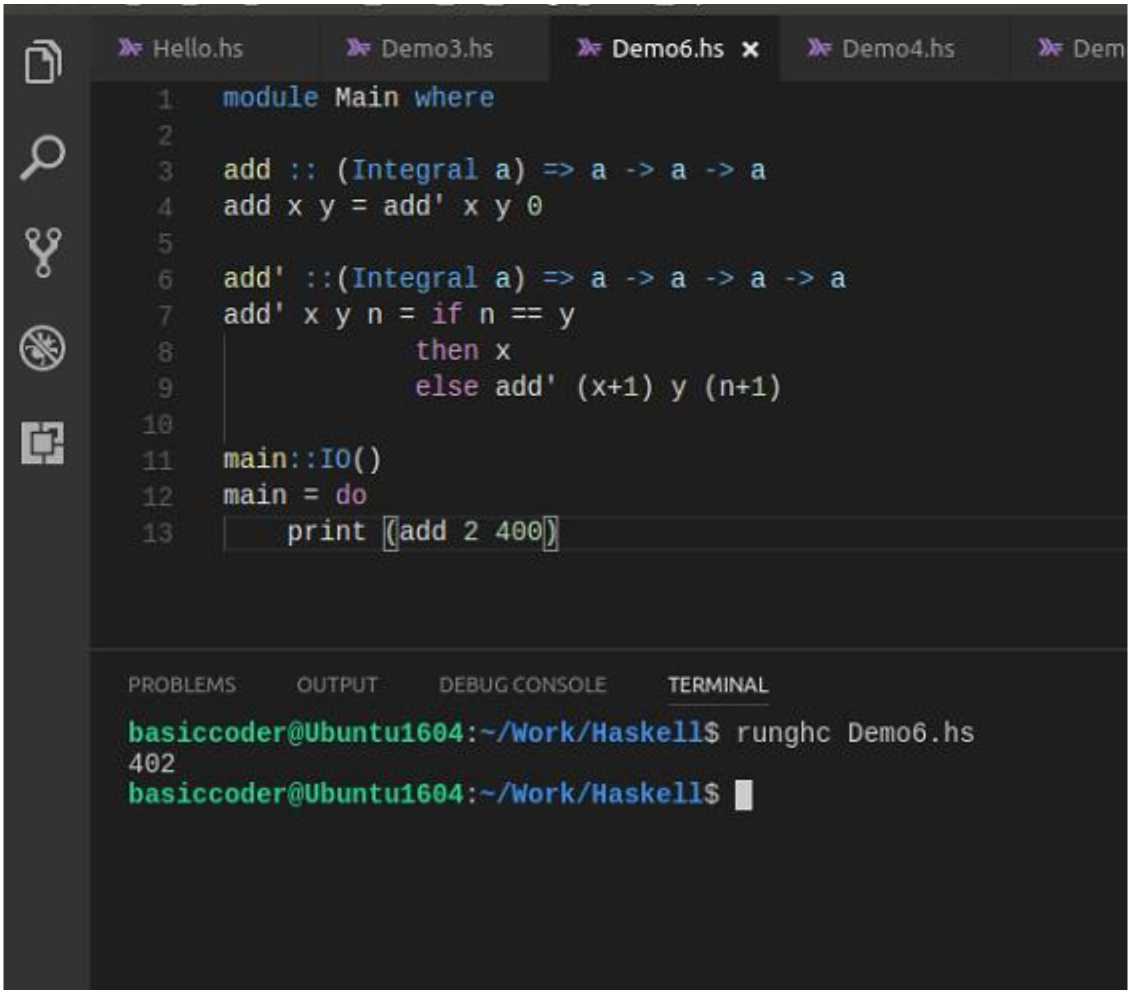
```
basiccoder@Ubuntu1604:~/Work/Haskell$ runghc Demo3.hs
6
basiccoder@Ubuntu1604:~/Work/Haskell$ 
```

二：实验结果：



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

三：实现(x+y) by 1+ 函数(add1)



```haskell
module Main where

add :: (Integral a) => a -> a -> a
add x y = add' x y 0

add' ::(Integral a) => a -> a -> a -> a
add' x y n = if n == y
             then x
             else add' (x+1) y (n+1)

main::IO()
main = do
    print (add 2 400)
```

```
basiccoder@Ubuntu1604:~/Work/Haskell$ runghc Demo6.hs
402
basiccoder@Ubuntu1604:~/Work/Haskell$ 
```

附件：

[1] Demo3.hs

[2] Demo6.hs