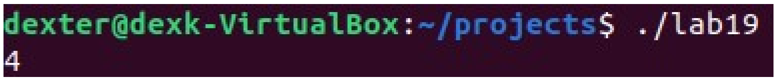
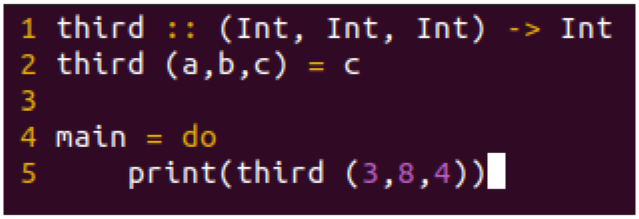
Lab 19
Dexter Kale – CS – 3035 – 05

Question 1:

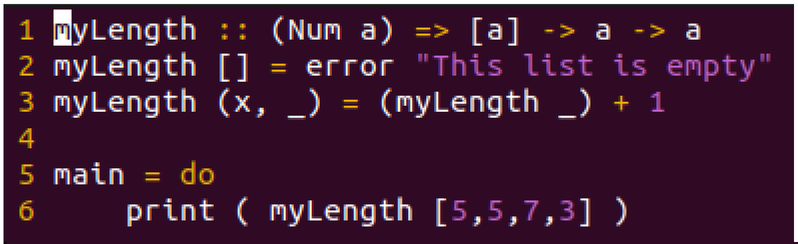


```
dexter@dexk-VirtualBox:~/projects$ ./lab19
4
```

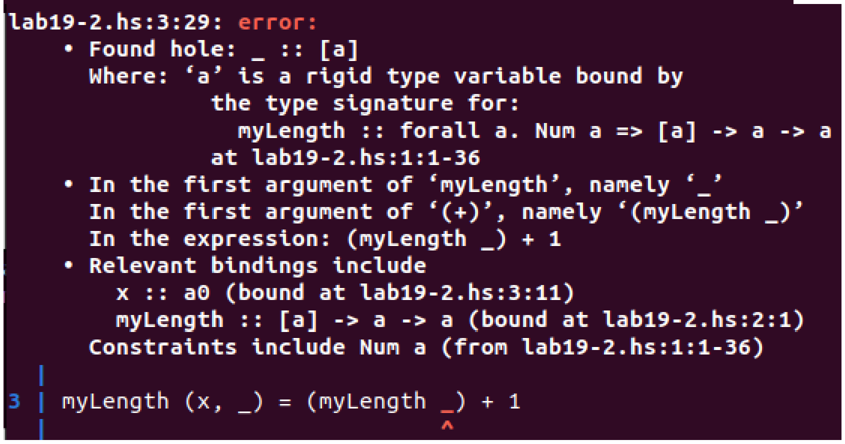


```haskell
third :: (Int, Int, Int) -> Int
third (a,b,c) = c

main = do
    print(third (3,8,4))
```

Question 2: I couldn't get this to work.

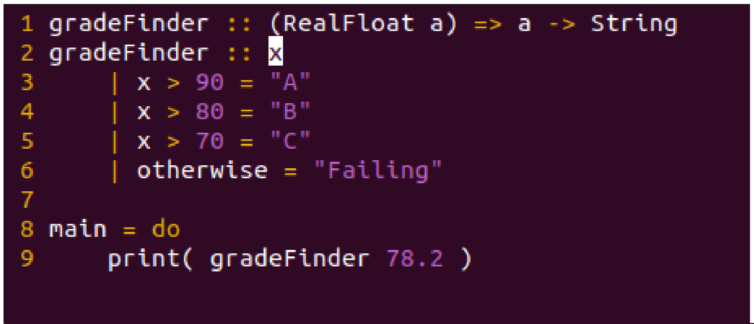


```haskell
myLength :: (Num a) => [a] -> a -> a
myLength [] = error "This list is empty"
myLength (x, _) = (myLength _) + 1

main = do
    print ( myLength [5,5,7,3] )
```

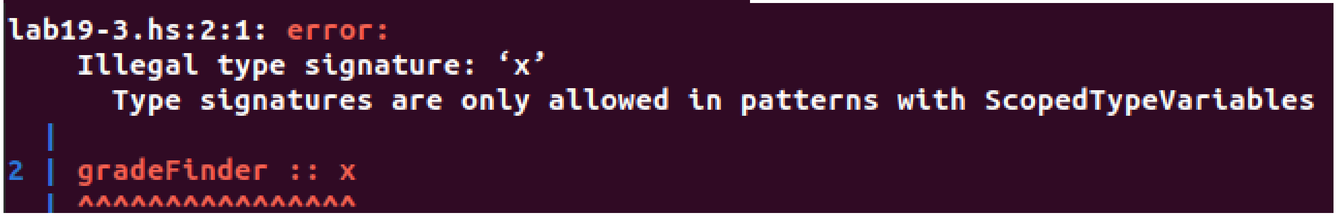


```
lab19-2.hs:3:29: error:
    • Found hole: _ :: [a]
      Where: ‘a’ is a rigid type variable bound by
               the type signature for:
                 myLength :: forall a. Num a => [a] -> a -> a
               at lab19-2.hs:1:1-36
    • In the first argument of ‘myLength’, namely ‘_’
      In the first argument of ‘(+)’, namely ‘(myLength _)’
      In the expression: (myLength _) + 1
    • Relevant bindings include
        x :: a0 (bound at lab19-2.hs:3:11)
        myLength :: [a] -> a -> a (bound at lab19-2.hs:2:1)
      Constraints include Num a (from lab19-2.hs:1:1-36)
  |
3 | myLength (x, _) = (myLength _) + 1
  |                             ^
```

Question 3: I couldn't get this to work.



```haskell
gradeFinder :: (RealFloat a) => a -> String
gradeFinder :: x
    | x > 90 = "A"
    | x > 80 = "B"
    | x > 70 = "C"
    | otherwise = "Failing"

main = do
    print( gradeFinder 78.2 )
```



```
lab19-3.hs:2:1: error:
    Illegal type signature: ‘x’
      Type signatures are only allowed in patterns with ScopedTypeVariables
  |
2 | gradeFinder :: x
  | ^^^^^^^^^^^^^^^^
```

Question 4:
In C, reassigning is easy and doable. In Haskell, not so much as most things are immutable.