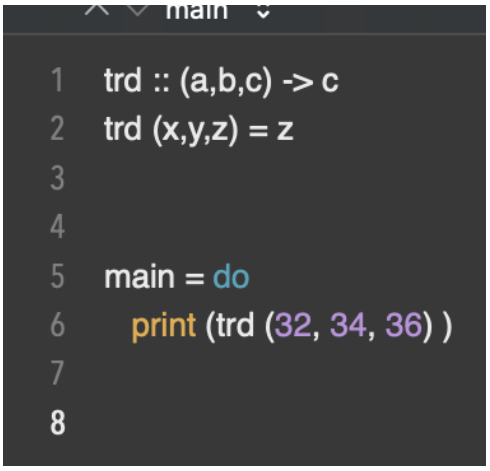
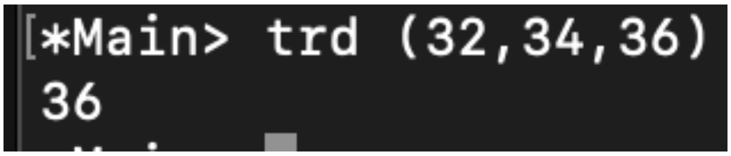
Part 1:

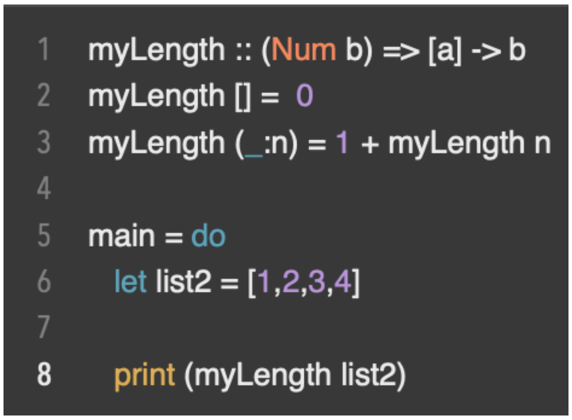


```
trd :: (a,b,c) -> c
trd (x,y,z) = z


main = do
  print (trd (32, 34, 36) )

```

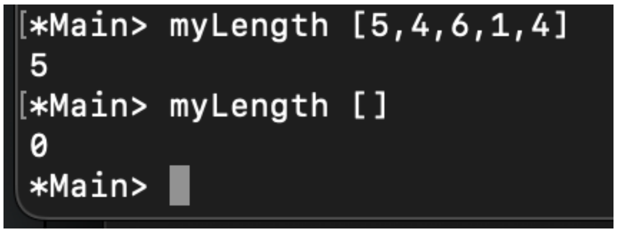


```
[*Main> trd (32,34,36)
36
```

Part 2:

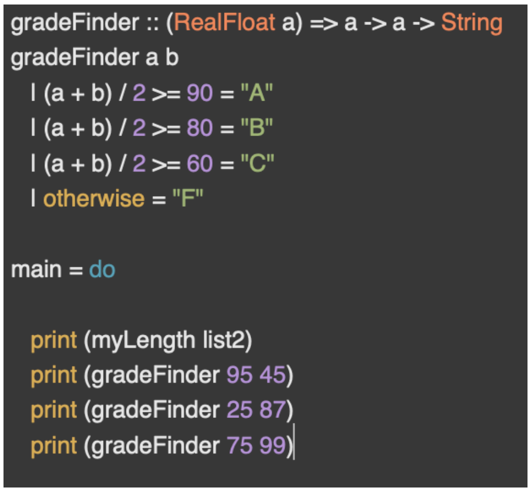


```
myLength :: (Num b) => [a] -> b
myLength [] =  0
myLength (_:n) = 1 + myLength n

main = do
  let list2 = [1,2,3,4]

  print (myLength list2)
```

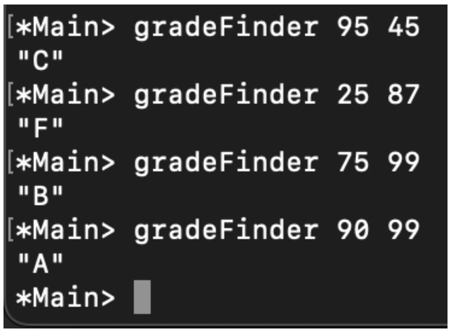


```
[*Main> myLength [5,4,6,1,4]
5
[*Main> myLength []
0
*Main> 
```

Part 3:



```
gradeFinder :: (RealFloat a) => a -> a -> String
gradeFinder a b
  | (a + b) / 2 >= 90 = "A"
  | (a + b) / 2 >= 80 = "B"
  | (a + b) / 2 >= 60 = "C"
  | otherwise = "F"

main = do

  print (myLength list2)
  print (gradeFinder 95 45)
  print (gradeFinder 25 87)
  print (gradeFinder 75 99)
```



```
[*Main> gradeFinder 95 45
"C"
[*Main> gradeFinder 25 87
"F"
[*Main> gradeFinder 75 99
"B"
[*Main> gradeFinder 90 99
"A"
*Main> 
```