```
takeFromList :: [a] -> Int -> a
takeFromList xs a = xs !! a
takeFromList [1,2,3,4,5] 3
```

```
*Main> takeFromList [1,2,3,4,5] 3
4
```

```
getArea :: Int -> Int -> Int
getArea x y = x * y
getArea 10 20
```

```
*Main> getArea 10 20
200
```

```
getList :: Int -> [Int]
getList a = [1..a]
getList 11
```

```
*Main> getList 11
[1,2,3,4,5,6,7,8,9,10,11]
```

```
isSqrtEq a b c = c == sqrt(a^2 + b^2)
isSqrtEq 4 5 6
```

```
*Main> isSqrtEq 4 5 6
False
```

```
isPythagorean a b c = isSqrtEq a b c || isSqrtEq a c b || isSqrtEq b c a
isPythagorean 3 4 5
```

```
*Main> isPythagorean 3 4 5
True
```

```
factorial n = product [1..n]
factorial 5
```

```
*Main> factorial 5
120
```

```
stooges n =  take n (cycle ["Moe", "Larry", "Shemp"])
slaps list n = ["slap " ++ stooge | stooge <- stooges n]
slaps (stooges 5) 5
```

```
*Main> slaps (stooges 5) 5
["slap Moe","slap Larry","slap Shemp","slap Moe","slap Larry"]
```