



A solution to the exercise

Exercise 1:

```
Cryptol> :s base=10
Cryptol> ["$"]
[[36]]
The answer is 36
```

Exercise 2:

Use the `/.` operator to get (ratio 238 81)

Exercise 3:

```
Cryptol> let x = [[[1,2],[1,2],[1,2:[32]]],[[3,4],[3,4],[3,4]],
                  [[3,4],[3,4],[3,4]],[[3,4],[3,4],[3,4]]]
Cryptol> :t x
x : [4][3][2][32]
```

Exercise 4:

```
Cryptol> (recip 37:Z 61)
33
```

Exercise 5:

```
Cryptol> (ratio (toInteger(-1)) (toInteger(2)))
(ratio -1 2)
or
Cryptol> (fromInteger(-1) /. fromInteger(2)):Rational
(ratio -1 2)
```

Exercise 6:

```
Cryptol> [2,4...]
[2, 4, 6, 8, 10, ...]
```

Exercise 7:

```
Cryptol> split `{4,2} [1,2,3,4,5,6,7,8]
[[1, 2], [3, 4], [5, 6], [7, 8]]
or
Cryptol> split [1,2,3,4,5,6,7,8]:[4][2][16]
[[1, 2], [3, 4], [5, 6], [7, 8]]
```

Observe the first of these is polymorphic on the number of bits to assign to the numbers in the sequence but the second of these is monomorphic and removing the `[16]` results in an error.

Exercise 8:

```
Cryptol> [True, True, False, True, False, True, False, True]:[8]
213
or
Cryptol> toInteger [True, True, False, True, False, True, False, True]
213
```

Exercise 9:

```
Cryptol> let f Q P = ~P \ / Q
```

```
Cryptol> let g = f False    // result of curry – what value of P makes g True
```

```
Cryptol> g False is True and g True is False
```

Developing a truth table works as well