

# Poker Hand

by ToF

July 29, 2014

$\lambda \ \lambda \ \lambda$

# 1 Values, Types and Functions

---

If we described the program we need to write as one function, what would be the name and type of this function?

1 `main :: IO()`

---

What would that function do?

2 Compute the scores extracted from the data in the standard input, and print them on the standard output.

---

How would you write this?

3

```
main :: IO()
main = interact computeScores
```

---

How would you write that function: `computeScores` ?

4

```
computeScores :: String → String
computeScores = unlines . mystery . lines
```

---

Your function breaks the input data into separate lines, processes all those line through a mysterious function, then assemble the result into a single output block of text. Why the name `mystery` ?

5 That's because I don't know what function I should write yet.

---

Let's try to specify it, then. What is the information we have about the function so far?

6 It consumes entry lines and yields those same lines with some score info added. So both input and output type are lists of `Strings`.

```
mystery :: [String] → [String]
```

---

How would you name a function that given a list of entry lines and a list of matching scores, would produce the output in the *mystery* function?

7

```
displayScores :: [String] → [Score] → [String]
```

---

How would you describe a score?

8 It's the combination of either the kind of hand the player has, or nothing if the player folded, and a value denoting if that player is the winner of the game or not:

```
type Score = (Maybe Kind, Bool)
```

---

What are values of the type named *Kind* ?

9

```
data Kind = HighCard
          | Pair
          | TwoPair
          | ThreeOfAKind
          | Straight
          | Flush
          | FullHouse
          | FourOfAKind
          | StraightFlush
          | RoyalFlush
```