**Haskell Lecture 2 演習問題**

**Maybeモナド**

readInts :: Int -> ByteString -> Maybe [Int]

readInts k s は、文字列sに空白区切りで書かれている整数をk個読み込み、それらのリストをJust付きで返す。k個未満しか数が得られない場合、Nothingを返す。

readInt :: ByteString -> Maybe (Int,ByteString) を用いて実装せよ。

readInts 3 “1 2 3 4 5” = Just [1,2,3]

readInts 7 “1 2 3 4 5” = Nothing

**Listモナド**

dice :: [Int] -> Int -> Int -> [Int]

dice [a1,...,ak] g n は、すごろくで、a1,...,akのk個の目が書かれたさいころをn回振ったあとにいる可能性のあるマスを重複なく昇順に列挙するリストを返す。

最初コマは0のマスにいる。

xのマスにいるときにaの目が出れば、x+aのマスに移動する。ただしx+aがg以上ならばすごろくをゴールしたことになり、gに移動し、その後はさいころの目にかかわらずgに留まり続ける。また、x+aが0未満ならば0に移動する。

**IOモナドの遅延評価**

machine :: String -> String

main = interact machine というふうにして使う。

この機械は内部に数字xを持っている。初期値は0である。

数字n を一行入力されると、何も表示せずxにnを足し、

sum と一行入力されると、“the sum is *x*” (*x*にxの値が入る) と一行表示し、xを0にリセットし、

end と一行入力されると、“goodbye” と一行表示し終了する。

**IOモナドで乱数生成**

乱数生成。

**モナド変換子 ― ListT IO**

getDirectoryContents :: FilePath -> IO [FilePath] を用いて何かを作る。

**RWSモナド**

インタラクティブなロボットを作る。

**Errorモナド**

簡単なパース。

**Contモナド**

手続き型っぽいことをする。

汎用関数を作らせる。

**Parsecモナド**

内部状態を持つパーサーを作る。

**Freeモナド**

複雑なモナドを作る。

プリティプリンタを作る。