

## 問題 9

○×ゲームを作りなさい。

これは、対戦ゲームなので、例えば次のようにゲームが進行する。(先攻は○とする)

- 1) 先攻が、記入位置を入力する
- 2) その位置に、○を記入した盤面を表示する
- 3) 後攻が、記入位置を入力する
- 4) その位置に、×を記入した盤面を表示する
- 5) どちらかが勝つか盤面が埋まるまで、1)~4) を繰り返す

---

```
*Main> gameStart
```

```
turn o > 1,1
```

```
---
```

```
-o-
```

```
---
```

```
turn x > 0,0
```

```
x--
```

```
-o-
```

```
---
```

```
turn o > 0,2
```

```
x-o
```

```
-o-
```

```
---
```

```
turn x > 1,0
```

```
x-o
```

```
xo-
```

```
---
```

```
turn o > 2,0
```

```
x-o
```

```
xo-
```

```
o--
```

```
Winner o
```

補足 1) 「○×ゲーム」とは 3x3 マスの盤に、○と×を交互に記入していき、○か×が縦横斜めのいずれかに3つ並べば、並んだ方の勝ち。

補足 2) 位置を入力する関数 入力については、getChar について簡単に説明しただけなので、○×を記入する位置を、入力するための関数の例を以下に示す。これをそのまま使っても良いし、参考にして自分で考えても良いことにする。

---

```
type Position = (Int, Int)

getInt :: Char -> Maybe Int
getInt '0' = Just 0
getInt '1' = Just 1
getInt '2' = Just 2
getInt _   = Nothing

makePosition :: String -> Maybe Position
makePosition [c1,s,c2] = do
    x <- getInt c1
    y <- getInt c2
    if s == ',' then Just (x, y) else Nothing
makePosition _ = Nothing

valid :: Position -> Bool
valid (x, y) = (0 <= x && x <= 2) && (0 <= y && y <= 2)

getPosition :: IO Position
getPosition = do putStr "please input: "
                 str <- getLine
                 let pos = maybe (-1,-1) id (makePosition str)
                     if valid pos then return pos else oncemore
                 where oncemore = putStrLn "once more" >> getPosition
```

---

(※) インデントに注意すること。特に関数 `getPosition` のところ。