

Solution Review: Devise Random Bit Generator

This lesson discusses the solution to the challenge given in the previous lesson.

```
package main
import (
    "fmt"
)

func main() {
    ch := make(chan int)
    // consumer:
    go func() {
        for {
            fmt.Print(<-ch, " ")
        }
    }()
    // producer:
    for i:=0; i<=100000; i++ {
        select {
            case ch <- 0:
            case ch <- 1:
        }
    }
}
```



Random Bit Generator

In the code above, at **line 7**, we make a channel **ch** for integers. Then at **line 9**, we start an *anonymous* function as goroutine: this continually gets values from the channel at **line 11**, in the infinite for-loop (from **line 10** to **line 12**).

The for-loop (from **line 15** to **line 20**), puts **100000** bits (0 or 1) on the channel. To do that, it uses a **select** (see **line 16**) and chooses randomly one of the case statements (either from **line 17** or **line 18**).

That is it about the solution. In the next lesson, there's another challenge for you to solve.

