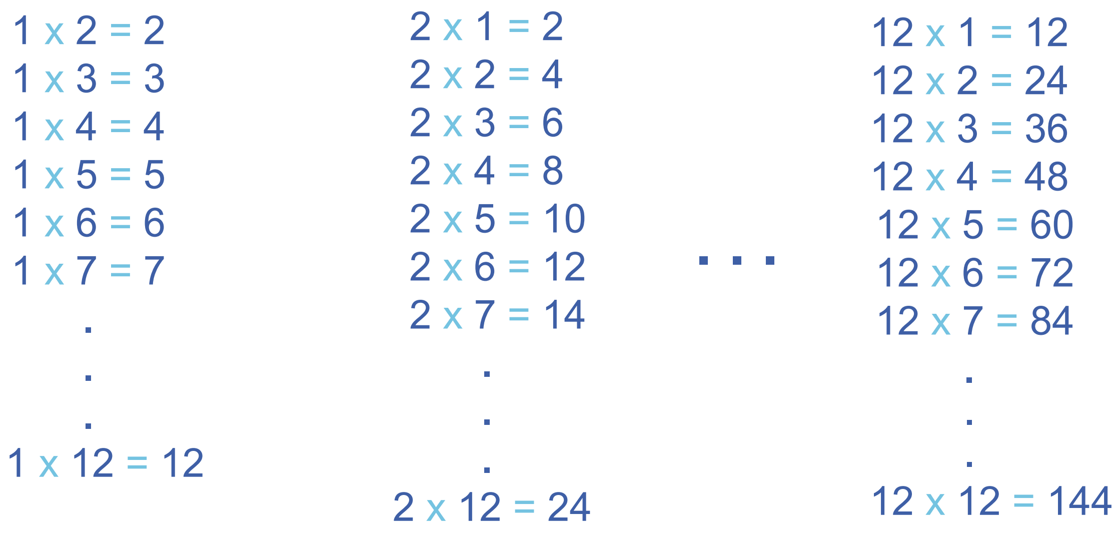
Exercise: Multiplication Table

Test your skills by printing out a multiplication table using goroutines and waitgroups!

In this exercise, you have been provided with a code which prints out the multiplication table from `1` to `12`. It creates goroutines in a for-loop which also run another for-loop for each number from `1` to `12`. However, nothing is printed out as the main routine exits before the goroutines are able to print.

You are required to solve this problem using `WaitGroup` from the `sync` package and output all the tables on to the console.



Multiplication Table

Now the output in the illustration above is very organized but since we are printing our table in goroutines which run concurrently, our output will be all over the place. This means that the following statement will execute in no specific order:

```
fmt.Printf("%d x %d = %d\n", i, n, n*i)
```

Here, you are required to maintain order with at least the variable `i`. For example, you can print all the calculations with the same value of `i` and then move on to the next value of `i`. Good Luck!

Current Output:

```
No output
```

Expected Output:

```
1 x ... = ...
1 x ... = ...
.
.
.
2 x ... = ...
.
.
.
12 x ... = ...
```

The second operand can vary and the answer of the multiplication depends on the second operand.

Show Hint

```go
package main
import "fmt"
func printTable(n int) {
  for i := 1; i <= 12; i++ {
    fmt.Printf("%d x %d = %d\n", i, n, n*i)
  }

}

func main() {
  for number := 2; number <= 12; number++ {
    go printTable(number)
  }
}
```

Multiplication Table

Hope you were able to solve the above challenge. Let's have a look at the solution review in the next lesson!