Challenge: Compute Factorial of a Number

This lesson brings you a challenge to solve.

WE'LL COVER THE FOLLOWING

Problem statement

Write a function that returns the **factorial** (!) of any `n` number. The factorial $n!$ of a number n is defined as:

n! = n * (n – 1)! , where 0!=1 and 1!=1

For example: $5! = 5 * 4 * 3 * 2 * 1 = 120$

Input

A number of type **uint64**, `n`

Output

A number of type **uint64** (because the number increases drastically due to multiplication)

Results will be correct until only `21!`. After that, the *overflow-error* will occur.

> **Remark**: When using type *int* the calculation is only correct up until `12!`. This is, of course, because an *int* can only contain integers that fit in

> 32 bits. And, Go doesn't usually warn against this overflow-error!

Sample input

```
10
```

Sample output

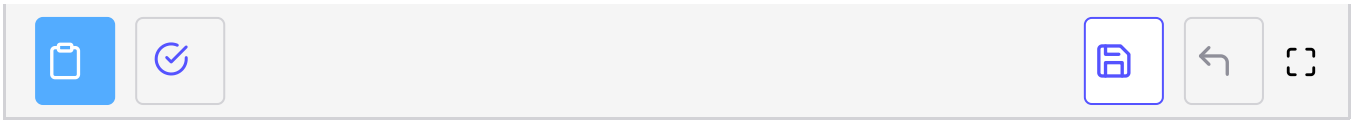
```
3628800
```

Try to implement the function below. Feel free to view the solution, after giving some shots. Good Luck!

```
package main
import "fmt"
import "strconv"
import "encoding/json"

func Factorial(n uint64) (fac uint64) {

        return
}
```



Factorial of a Number

We hope that you were able to solve the challenge. The next lesson brings you the solution to this challenge.