Certainly, let's tackle these Go programming challenges one by one:

1. WAP in go to print table of given number:

```
package main

import "fmt"

func main() {
    var number int
    fmt.Print("Enter a number: ")
    fmt.Scan(&number)

    fmt.Println("Table of", number)
    for i := 1; i <= 10; i++ {
        fmt.Printf("%d x %d = %d\n", number, i, number*i)
    }
}
```

2. WAP in go language to print PASCALS triangle:

```
package main

import "fmt"

func factorial(n int) int {
    if n == 0 {
        return 1
    }
    return n * factorial(n-1)
}

func combinations(n, k int) int {
    return factorial(n) / (factorial(k) * factorial(n-k))
}

func printPascalTriangle(rows int) {
    for i := 0; i < rows; i++ {
        for j := 0; j <= i; j++ {
            fmt.Printf("%d ", combinations(i, j))
        }
        fmt.Println()
    }
}

func main() {
    rows := 5
    fmt.Println("Pascal's Triangle:")
    printPascalTriangle(rows)
}
```

3. WAP in go language to print Fibonacci series of n terms:

```
package main

import "fmt"

func fibonacci(n int) int {
    if n <= 1 {
        return n
    }
    return fibonacci(n-1) + fibonacci(n-2)
}

func main() {
    n := 10
    fmt.Print("Fibonacci Series: ")
    for i := 0; i < n; i++ {
        fmt.Print(fibonacci(i), " ")
    }
    fmt.Println()
}
```

4. WAP in go language to illustrate pointer to pointer concept:

```
package main

import "fmt"

func main() {
    var x int = 10
    var p *int = &x   // p is a pointer to x
    var pp **int = &p // pp is a pointer to p

    fmt.Println("Value of x:", x)
    fmt.Println("Address of x:", &x)
    fmt.Println("Value of p (address of x):", p)
    fmt.Println("Address of p:", &p)
    fmt.Println("Value of pp (address of p):", pp)
}
```

5. WAP in go language to explain new function:

```
package main

import "fmt"

func main() {
    // Allocate memory for an integer using new
    var ptr *int = new(int)
    *ptr = 42
```

```
    fmt.Println("Value stored in the allocated memory:", *ptr)
}
```

These programs cover the requested concepts in Go. Feel free to ask if you have any further questions or want to explore specific aspects in more detail!

- https://juejin.cn/post/7271499307373723706