**Task 0:** Can you write a Golang script that uses two goroutines to print odd and even numbers, with one routine handling even numbers and the other handling odd numbers?

**Task 1:** How can one determine if a given integer is a power of 2 without using the pow function from the math package or loops?

**Task 2:** Given a set of non-negative integers, how can one find the highest amount sequences, with a sequence defined as a series where each number differs by 1 from the next? The numbers in the sequence do not have to be in sequence within the original set.

**Task 3:** How can one shift a given set of numbers by a specified index? For example: [5,7,8,4,9,1] shifted by an index of 3 results in [9,1,5,7,8,4]. This should be performed "in-place" without creating an additional array.

**Task 4:** Given an ordered set of numbers, how can one re-sort it so it alternates between maximum and minimum values (first maximum, first minimum, second maximum, second minimum, and so on)? Can this operation be performed "in-place" without creating an additional array? For example: [2,3,5,7,8,9] -> [9,2,8,3,7,5].

**Task 5:** Given a string expression containing numbers and arithmetic operations (only '+' and '-'), how can one calculate the value of the expression without using ready-made parsers? For example: "2+6+4" -> 12, "20-10-6" -> 4, "123+200" -> 323.

**Task 6:** How can one pass a given number to a goroutine, double it, pass it back to the main, and print the result? Please provide an explanation of the implementation and possible solutions.

**Task 7:** Can you explain what the following Golang code does, if it compiles and possible outputs?

**Task 8:** Could you write a Go program that uses two goroutines to generate random numbers and print the count of these numbers - one routine prints the number, the other prints the count of numbers generated?

**Task 9**: Can you create a 'Ping-Pong' game using channels and goroutines in Go? There are two players and the game goes on for two minutes. Each player serves the ball after waiting for a random period (between 0 and 10 seconds) and the player holding the ball at the end of two minutes wins. The game needs to use context and goroutines.

**Task 10:**

func main() {

    m := make(map[string]string)

    m["a.b.c.d"] = "123"

    m["a.b.e.f"] = "456"

    m["a.g.h.i"] = "456"

    m["w.x.y.z"] = "345"

}

/\*

Output

    a:

        b:

            c.d:123

            e.f:456

        g.h.i: 456

    w.x.y.z: 345

\*/