Build Instruction:

1. Define The variables ( $array for keep values and $col for number of values )
2. Randomly select first number.
3. Then start to fill the 9 other positions randomly.
4. When any position filled, it will start checking the duplication with other positions.
5. If duplication occurs, it will change the value and start checking from beginning.
6. This loop will repeat until all positions fill and each value appear only once.

Usage :

Random number generators have applications in gambling, statistical sampling, computer simulation, cryptography, completely randomized design, and other areas where producing an unpredictable result is desirable. Generally, in applications having unpredictability as the paramount, such as in security applications, hardware generators are generally preferred over pseudo-random algorithms, where feasible.

Random number generators are very useful in developing Monte Carlo-method simulations, as debugging is facilitated by the ability to run the same sequence of random numbers again by starting from the same random seed. They are also used in cryptography – so long as the seed is secret. Sender and receiver can generate the same set of numbers automatically to use as keys.

Random numbers are very useful for security reason, For example any HTTPS session starts as follows:

1. The web browser sends information to the server about which version of SSL it wants to use and other information.
2. The web server replies with similar information about SSL versions and its SSL certificate.
3. The web browser checks that the certificate is valid. If it is, it generates a random 'pre-master secret' that will be used to secure the connection.

After that further exchanges occur all based on the randomly chosen pre-master secret. It needs to be unpredictable for the connection to be secure.

Description :

This script can randomly produce values from 1-10 without duplication.

Known limitations / bugs :

Limitation is for extending this script. Because it uses RANDOM function in bash and this function only produce 0 to 32767. Another limitation is high CPU usage in big values, because it will check all the values to prevent duplication. And it’s not useful for create random values bigger than 32767.

PS: Many people can write a code that computer can understand, but a real engineer is who writes a code that every human can understand. ☺