Tasks to do:

1. create a public GIT
2. Build an encoder that encodes the content of a defined structure into a byte stream (network-byte-order)
3. Use the external GO libraries provided for this purpose under
   -
   https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3370
   - https://github.com/campusgeniuspub/structex
4. Create unit tests that test your own implementation.
5. Use code coverage tests to check the coverage of your own unit tests.

Help:

Here is a GO tour that makes it easier to get started with the programming language
https://tour.golang.org/welcome/1

Creating a public GIT:

Verwende dafür https://github.com
Make it public, so you can share your results

Building an Encoder:

The encoder is to convert an information element (IE) into a byte stream.

The IE to convert (encode) shall be:

- 9.11.3.9A (5GS Update Type) in 3GPP TS 24501 (Version 16.9.0)
  https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3370
- The IE consists of 3 octets (Bytes), the last octet consists of a bitfield
- To convert a bitfield use structex (More information can be found here:
  https://github.com/campusgeniuspub/structex)
- A struct for the IE Nas5GSUpdateType shall be created. The bitfield tagging of structex must be used
- Create a function that converts any Nas5GSUpdateType object into a byte stream. The function header should look like this:
  func (ie Nas5GSUpdateType) Encode(buffer *bytes.Buffer)

- bytes.Buffer: The Buffer can be used by importing "bytes".

Unit-Test:

Create a Unit-Test, which tests and verifies your code. Use the following testvector

Please use the structex from https://github.com/campusgeniuspub/structex for the test

Input:
Nas5GSUpdateType {
IEI=1
Length=2
EPS-PNB-CIoT=0
5GS-PNB-CIoT=0
NG-RAN-RCU=1
SMS-requested=1
}
Output:
Bytestrom=0x01, 0x02, 0x03

Code-Coverage:

Create and document code coverage for your own code. Achieving 100% is not necessary.