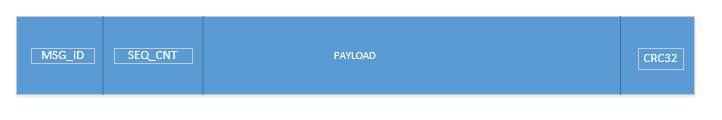
1. Create a Client application running on LocalHost that connects at startup on port 55000 to the Server on port 55000
2. The client should be able to receive from the server the following data frame of 512 bytes:

* MSG\_ID, SEQ\_CNT and CRC32 are uint32 types
* PAYLOAD is an array of uint8[500];



1. The client should be able to store a “synchronized package” of 3 frames (IDs: 1, 2 and 3) based on the SEQ\_CNT

A Synchronized package consist of 3 frames that have incrementing IDs ( 1, 2, 3) and the same SEQ\_CNT (eg: 0, 1, 2...)



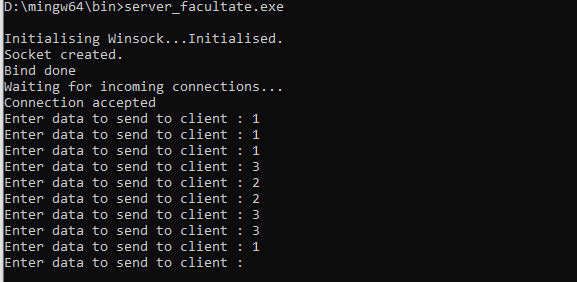
1. The client should be able to store the last 3 synchronized packages, in a circular buffer. The highest SEQ\_CNT has the highest priority

* Use the provided Server application, that listens on Port 55000 for any incoming connection. The server will send the data frame with incremental SEQ\_CNT for each transmission at on “key press” events as follows:
  + “1” -> data frame with MSG\_ID = 1
  + “2” -> data frame with MSG\_ID = 2
  + “3” -> data frame with MSG\_ID = 3

Eg: On the first key entered as “1” it will send a package with MSG\_ID=1 and SEQ\_CNT=0

On the second key entered as “1” it will send a package with MSG\_ID=1 and SEQ\_CNT=1

Etc..



Package sent for the input given above:

- MSG\_ID=1, SEQ\_CNT=0, Payload[500], CRC

- MSG\_ID=1, SEQ\_CNT=1, Payload[500], CRC

- MSG\_ID=1, SEQ\_CNT=2, Payload[500], CRC First synchronized package

- MSG\_ID=3, SEQ\_CNT=0, Payload[500], CRC

- MSG\_ID=2, SEQ\_CNT=0, Payload[500], CRC

- MSG\_ID=2, SEQ\_CNT=1, Payload[500], CRC Second synchronized package

- MSG\_ID=3, SEQ\_CNT=1, Payload[500], CRC

- MSG\_ID=3, SEQ\_CNT=2, Payload[500], CRC

- MSG\_ID=1, SEQ\_CNT=4, Payload[500], CRC

Evaluation:

1. 1p
2. 2p
3. 2p
4. 1p