Exercise 1 – readme file:

205892201 Guy Haviv – [guyhaviv@mail.tau.ac.il](mailto:guyhaviv@mail.tau.ac.il)

316294636 Noa Davidovitch – noad1@mail.tau.ac.il

This is an implementation of a Server/Client protocol for the game "nim" as described in the exercise description.

Files & Modules:

2 files only: nim-server.py for server application, nim-client.py for client application.

Run the code:

1. Establish a Server – python3 nim-server.py n\_a n\_b n\_c [,PORT = 6444]
2. Run a client – python3 nim-client.py [,HOSTNAME = localhost,PORT = 6444]

Protocol:

The protocol is based on a two-way **BINARY** communication between client & server in the following message construction (format for struct.pack/unpack) :



Client sends a heap identifier (A/B/C) and amount of die to remove from that heap.

When a Client user inputs 'Q' , the client program quits, and server will follow on data receive error.

Server sends a message tag (char) that stands for the game status (which can rely on the Client's previous move) and the amount of die in each heap.

Errors and edge cases:

This is the program's error handling policy for both server & client:



\* Client: if *no program arguments* passed – connects to localhost:6444, if *one argument* is passed it will be considered as HOSTNAME , i.e. connects to HOSTNAME:6444.

For 2 arguments or more – will take the *first two* as HOSTNAME:PORT.

Server: program arguments *must* include at least 3 integers for heap sizes, optional 4th argument as PORT .

5 arguments or more , 2 arguments or less - will result in error.

Logically invalid arguments (e.g. a string 'hello' as heap size for server) will result in error.