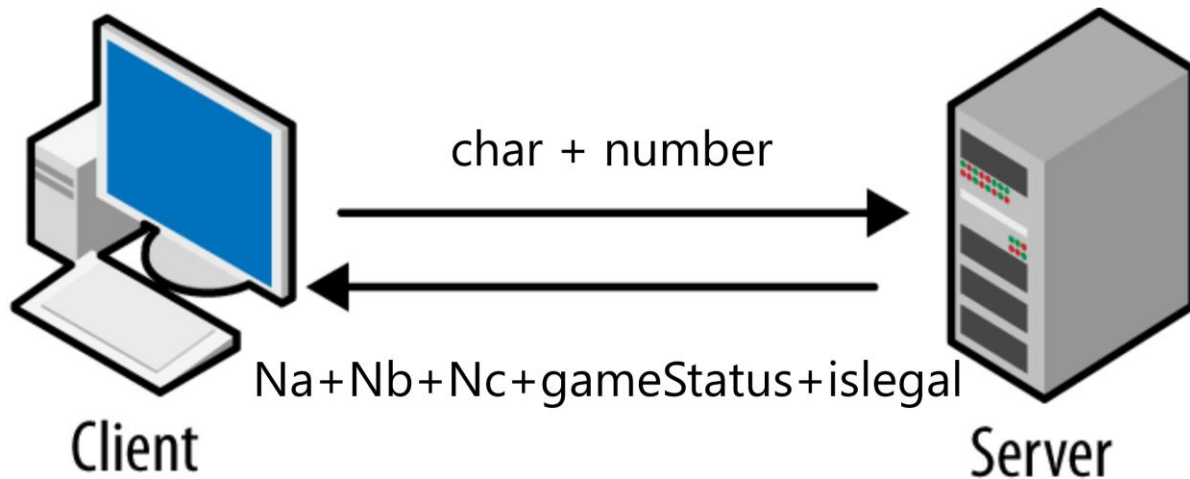


yazanzoabi 209105790 [yazanzoabi@mail.tau.ac.il](mailto:yazanzoabi@mail.tau.ac.il)

khimel 207001702 [khimel@mail.tau.ac.il](mailto:khimel@mail.tau.ac.il)

Protocol:



**Server** sends 5 short variables in a struct pack, the first 3 of them are the updated 3 values of the heaps A, B and C.

the fourth variable represents the game's status: 1 means server won, 2 means client won and 3 means the game ended with the input Q, any other value will be ignored and the game will continue normally.

The fifth value represents the legal checker's output, meaning if the player's input was legal this value will be 1 if it was illegal this value will be 2. Other values will be ignored(player didn't play yet or game was over).

Prints on client's side will be decided by these variables.

**Client** sends one char and one short value.

In case of legal input by the player(minimal input check on the clients side is done) this char will be A, B or C to represent the relevant heap. Z for quitting the game(Q) and D for illegal input(again minimal checking is done on the client's side).

The short value represents the amount requested to be subtracted by the player.

## Quick explanations

**Files:** Our submitted file has 3 python programs. Nim-server is the server's side, nim is the client's side and MySocketClass is a class that is used for both and must be in the folder with both or with each of the other files.

**nim:** client's program, responsible for the following:

communication: establishes connection with a running server program, receives and sends data.

Game control: waits for input from the player and checks that It's made out of two phrases separated by a blank " ". checks if the first phrase is A, B or C and that the second one is a number(sends the legal input), or that the input is simply "Q" in that case it sends Z for quitting, otherwise will send "D"(illegal move).

Closes socket in case the game is over.

**nim-server:** server's program, responsible for the following:

communication: accepts a connection started by a running client program.

Additional legal check: in case the char received is A, B or C checks that the short value is lower or equal to the number in the relevant heap, if the char is "Z"(Q) it ends the game if it's "D"(illegal) it skips the player's try.

Game: in case of a legal input by the player it calculates the new values for the heaps then checks if the player won if not the server plays and then checks again if it won. Then it sends 5 short values as explained above.

Closes socket in case the game is over. But keeps running for new games.

**MySocketClass:** is a helping class for easier communication between the client and the server.

- \* checks for connection errors

- \* implements a new recv and send functions that deal with communication errors and corruption.

This class is programmed for sending and receiving data using fixed lengths that can be customized. And makes sure the sent data is received in a complete form.