Name: Bar Yaacovi
ID: 208939009
Mail: baryaacovi@mail.tau.ac.il

**********

~~~~Application protocol~~~~

Before every string sent by both sides, they send the length
of it so the other side knows how many bytes to read into his buffer.

When the client understands what function is required, he sends the op code to the server.
The functions are labeled 0 to 4:
0. list_of_files
1. delete_file
2. add_file
3. get_file
4. quit

The client sends a code to the server represeting the function.

The server sends 1 for success and -1 for failure
if it is required:
In delete_file and add_file, this is the last message.
In case of failure in get_file it is sent instead of the length of the required file content.

Both client and server need to use the same constants from the file: "constants.h".

Run the programs like explained in the excercise page:
file_client [hostname [port]]
file_server users_file dir_path [port]

The program is divided into 3 files:
constants.h - contains constants both client and server have to work with.
client.c - the program that runs the client, works by the described protocol
above.

**********

Assumptions:
* The user_dir paramter passed to the server
ends with '/'
* if the client inserts wrong ID and/or pass,
send an error message, and let him try again.

* If an error occurs in delete \insert \getfile, tell the client, and
allow to insert a new function.

* add_file - if the file already exists on the server then do
nothing and have client print "file already exists".
~~~~

* get_file - if the file exists on my computer already, then overwrite it.

* get_file - the path is of the directory, name has to be same as on the server.

* get_file - assuming the dir path ends with '/'.

* incorrect format func - go to read again

* error like memory alloc - just write to the side that has error.

* if the maximum files amount for a user has been reached - won't let insert.