README
TCP Socket Programming

Files:
Client.c
Server.c
Helper.c

Functions:
Client->
In my program, first in order to get command line arguments, I created temporary char variables and stored the command line arguments therein. Now I had to send those command line arguments, where information like filename, file length and file itself were to be sent.

In order to send all those information, I first created a buffer where I stored format type in char. Similarly, the name of the file was decided by client so filename itself was to be sent to the server. So to send the file name, I first find out the length of the filename and stored in some array and sent the file name as well as its length to the server. Later in the server, I took the length of the filename and defined a character array with the length of the filename. I also sent the length of the file and created the character array in the server. After sending the length of the file, I also sent my file array inside the buffer where I stored other information in. With that I sent the file, file path, target file name and file itself.

Server->
After I write all my files and other information on the socket connection, I read the data from the socket in the server. I read the format option and if the format option was not correct I sent out an error message back to the client. I read the target file length, file length and the file itself. After retrieving all the information from the socket, I sent file, file length, target file name and format option back to a helper function, which processed the files and converted it based on appropriate format based on format option.
After the file processing and writing to a target file name was completed, I sent a confirmation message back to the client.

Helper-> Helper function received a pointer to a file buffer, file length, pointer to convert option and target filename. I used a number of functions inside the helper function to facilitate our read and write files scenario.
void process_file(char* buffer, int length_of_file, FILE* pointer);
int typezero_input(int pos, char* binary_buffer, FILE* pointer, char* convert_option);
void convert_to_typeOne(FILE* pointer, uint8_t amount, uint16_t numbers[]);
int typefirst_input(int pos, char* binary_buffer, FILE *pointer, char* convert_option);
void convert_to_typeZero(FILE* pointer, uint8_t amount, short numbers[]);

Github link: https://github.com/idiosincrasia/Network_Project1.git/': Could not resolve host: github.com