Deadline is same as the Assignment 2

Additional assignment for Lab 02 (for bonus)

File transfer and receiving using UDP sockets

UDP client for file transfer

Write a UDP client program that does followings:

1) Opens the **"innopolis.jpg"** or other image file in binary mode and reads the contents
2) Calculates its CRC checksum using given **get_crc_checksum()** function

```
import binascii

def get_crc_checksum(file_contents):

    file_contents = (binascii.crc32(file_contents) & 0xFFFFFFFF)

    return "%08X" % file_contents
```

3) Creates a dict called **file_info** with "checksum", "name", and "size" keywords and their values: checksum, name of the file, and size of the file respectively.
4) Creates a UDP socket
5) Sends the dict to the server. You need to use json to serialize the dict into a string then you can encode and send the string. Ex)

```
import json

string = json.dumps(file_info)

s.sendto(string.encode(), (ser_ip_addr, ser_port))
```

6) Then waits to receive a **buffer_size** info from server during 1 second. The server sends a string representation of its **buffer_size**.
    - If the **buffer_size** isn't received within a timeout, the client should print "The server isn't available" and shut down.
7) Then sends the contents of the file as follows: using a loop, send **buffer_size** bytes each iteration till there is no bytes remaining. You don't need to encode the file contents before sending since it's already bytes object.
8) Then waits for an "OK" message from server during 1 sec.
    - If the message is received, then it means the file was sent correctly. And the client app shuts down
    - Otherwise, repeat everything from Step 5

UDP server to receive the file

Your server must always be ready to receive the files from clients.

1) Receives the file_info from the client, decodes it, and then converts to dict using json.loads(string). The file_info should be a dict such as:

    Ex) file_info = {'checksum': '97F6DB4F', 'size': 65066, 'name': 'some_image.jpg'}

Deadline is same as the Assignment 2

- file_info['checksum'] is the checksum of the file to be received and calculated by the client using get_crc_checksum() function
- file_info['size'] is the size of the file_contents in bytes
- file_info['name'] is the name of the file that will be sent by the client

if there is any exception while performing above operations, your program should catch it and send it to the client. Your program shouldn't stop but go back to step 1 and wait for a new file.

2) Then prints the client address (i.e., ip and port #) and dict it just received (look at the example capture below)
3) Then sends its buffer_size to the client. Set the buffer_size small value like 100.
4) Then waits to receive the file contents inside the infinite while loop.
5) After it receives all file_info['size'] bytes, it calculates the checksum for file_contents it received.
    - If the checksum is the same as file_info["checksum"] open a new file called as "new_"+file_info["name"] in write-binary mode and write the received file_contents, and sends "OK" to the client informing about successful reception of the file, and go back to step 1 and continue waiting for a new file
    - Otherwise, go back to step 1 and continue waiting for a new file