Zixuan Cui (Elle) & Vladyslav Polishchuk (Vlad)

3/7/2019

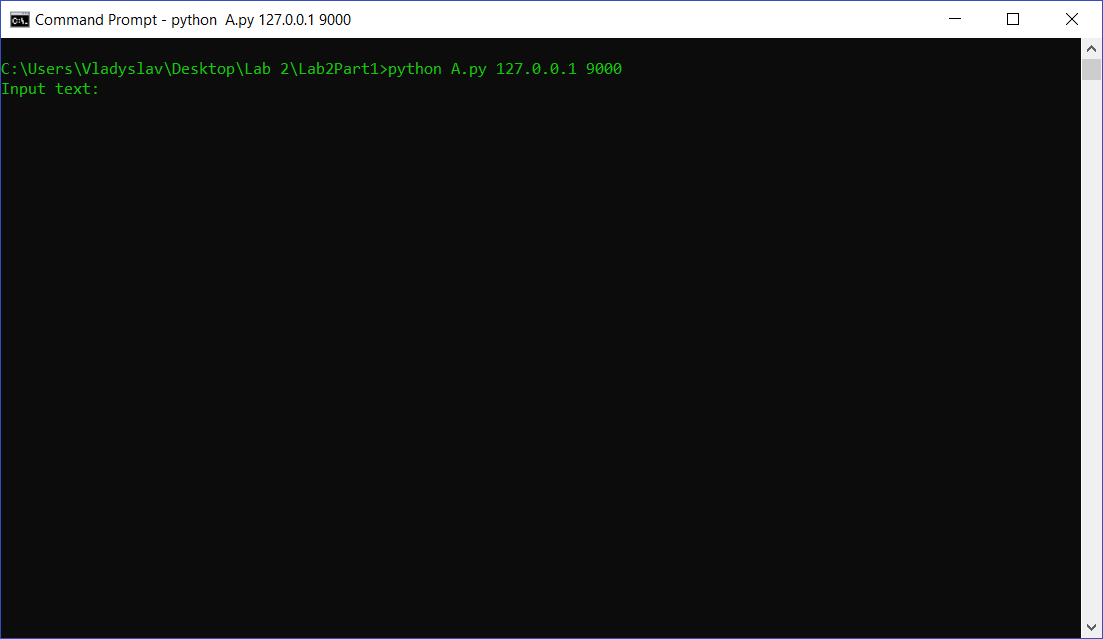
Data Communications

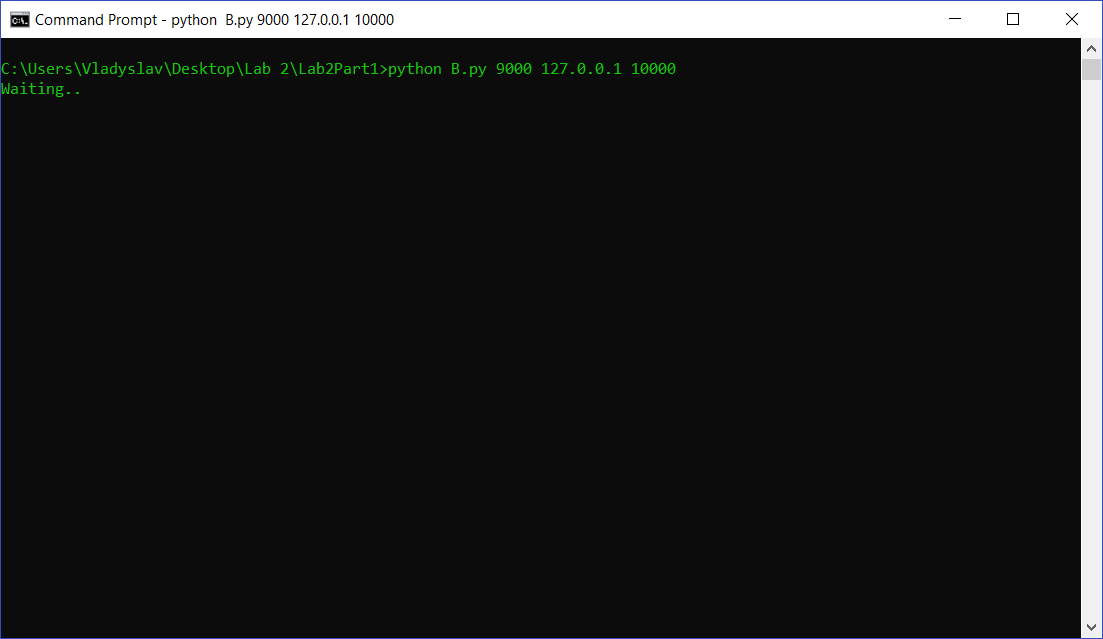
Professor Ying Mao

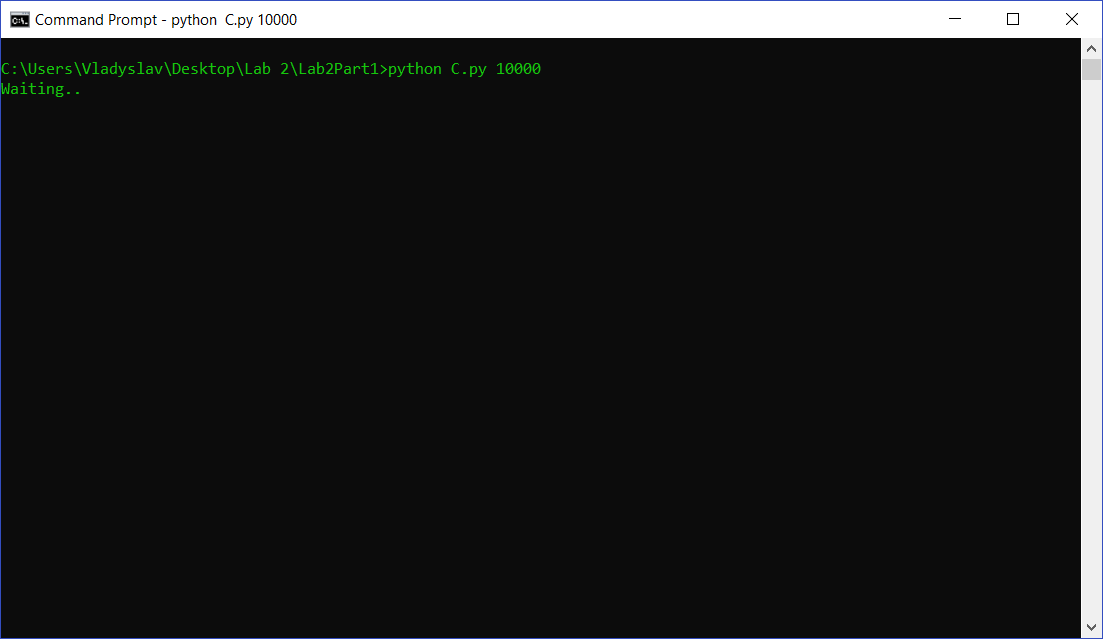
Lab 2 Part 1 Report

* It took us a while to figure out the structure of part 1 – we initially only considered the possibility of having two files. However, the problem we obviously ran into was figuring out exactly how file B would be able to be both a forwarder and a receiver. It’s only after you clarified the need for 3 separate files that everything fell into place.
* File A.py was made an unchanged copy of client.py from the sample-1 provided with our lab instructions. File C.py was made an unchanged copy of server.py (also from sample-1). File B.py, the extra file we realized had to be created, took on elements of both A and C. With A mirroring a client (sending) and C mirroring a server (receiving), B had to include the functionality of both (sending and receiving).
* Several changes had to be made to B to make the above plan work: the number of arguments that B needed would be greater than the ones expect by A (3) or C(2). Since B was acting as a forwarder, it required a listening port, an IP, and a sending port (with the fourth argument being the default sys.argv[0]). This means that “if len(sys.arv) !=” had to be set to 4. Additionally, a second socket had to be bound since B was forwarding its message onto C. This meant that variable “s” from A was renamed to “s1”, and a second socket variable “s2” was created. Data was received from s1 (“data, addr = s1.recvfrom(1024)”) and sent through s2 (“s2.sendto(data.encode(),…”).

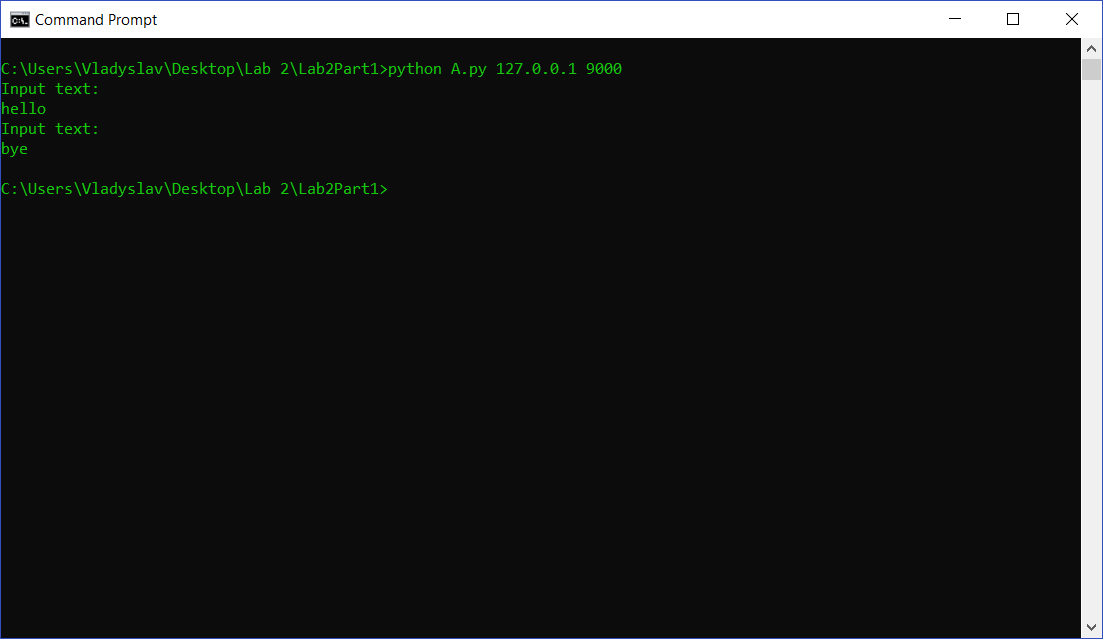
**Pre:**

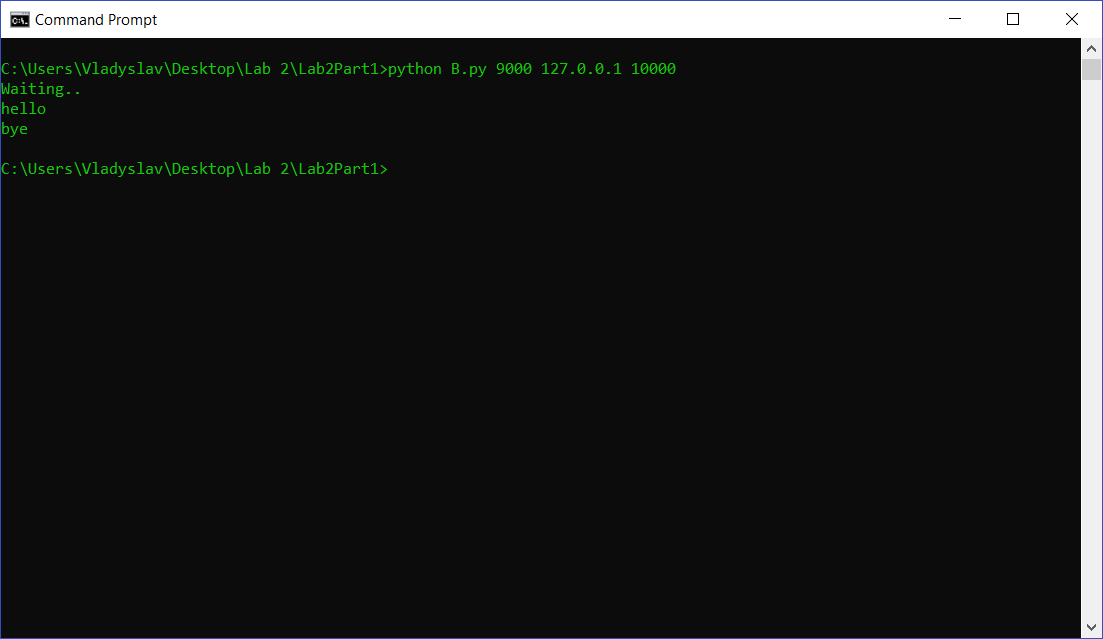
****

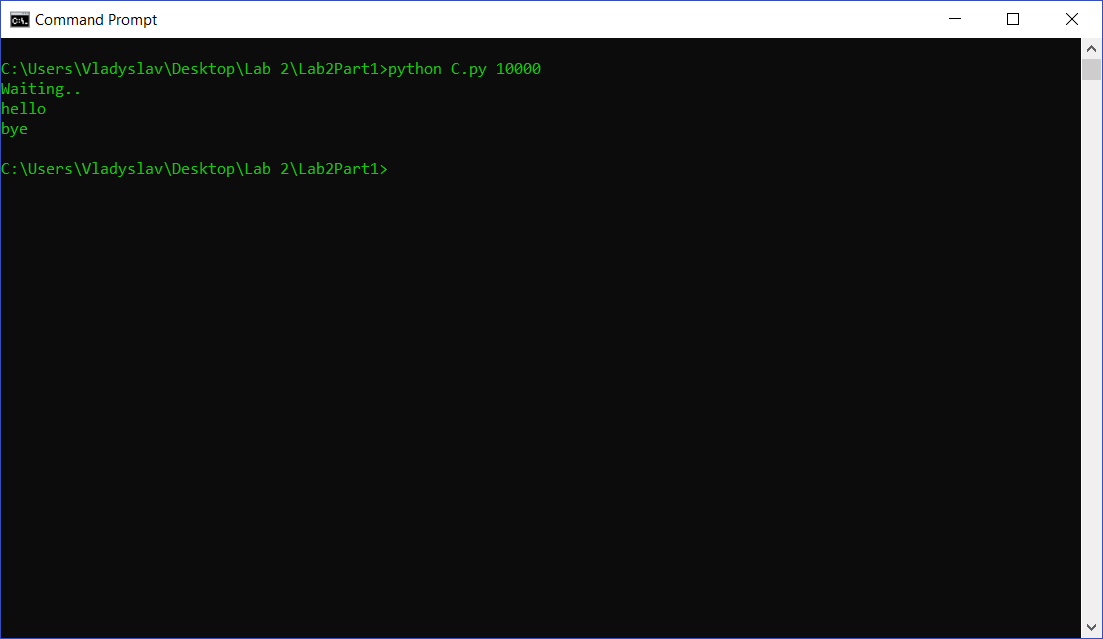
****

****

**Post:**

****

****

****