



به موارد زیر توجه کنید:

- ۱- برنامه‌های زیر را ترجیحا به زبان پایتون، اگر نه به زبان جاوا یا C++ بنویسید.
- ۲- نام و شماره دانشجویی خود را در بالای همه برنامه‌ها بنویسید
- ۳- کارکرد بخش‌های اصلی برنامه‌ها را با یک کامنت مناسب توضیح دهید.
- ۴- متن برنامه‌های هر سوال را با فرمت zip در سامانه کوئرا آپلود کنید.
- ۵- برنامه زمان‌بندی تحویل آنلاین متعاقبا اعلام خواهد شد.
- ۶- این تمرین ۲۰۰ نمره دارد که معادل دو نمره از ۲۰ نمره نهایی درس است.
- ۷- در صورت مشاهده هر گونه مشابهت نامتعارف کل نمره این تمرین را از دست خواهید داد.

در نوشتن برنامه‌ها به این نکات هم توجه کنید:

- در تمامی برنامه‌ها باید همه پیام‌هایی که تحت شبکه ارسال می‌شوند توسط فرستنده و گیرنده چاپ شوند. فرمت این لاگ‌ها برعهده خودتان است اما در لاگ چاپ شده باید زمان ارسال/دریافت پیام با ساختاری که برای انسان قابل خواندن باشد (با دقت میلی‌ثانیه) اعلام شود. مثال:

22:26:12:54 PM "sending request to port 8989"

- با توجه به اینکه کدهای شما به صورت محلی (local) اجرا می‌شوند، تاخیر شبکه بسیار ناچیز خواهد بود. در صورت نیاز برای شبیه‌سازی این تاخیر می‌توانید از تابع sleep در زبان موردنظر خود استفاده کنید.

سوال اول (۸۰ نمره):

- ۱- در این سوال قصد پیاده‌سازی یک UDP Pinger داریم. شما باید هر دو برنامه سرور و کلاینت را پیاده‌سازی کنید. کلاینت ۱۰ پینگ را به سرور ارسال می‌کند. چون UDP به صورت unreliable است، بسته‌های ارسال شده ممکن است در شبکه گم شوند. به همین دلیل کلاینت بعد از ارسال پینگ لزوما پاسخ آن را دریافت نمی‌کند. در نتیجه شما باید کلاینت را به گونه‌ای پیاده‌سازی کنید که پس از مدتی (مثلا دو ثانیه) در صورت دریافت نکردن پاسخ از سرور فرض کند که این بسته در شبکه گم شده است. از طرفی سرور را هم باید به گونه‌ای طراحی کنید که پیام‌های پینگ دریافت شده از کلاینت را uppercase کند و به کلاینت ارسال کند. همچنین انتظار می‌رود ۳۰ درصد پکت‌ها در شبکه گم شوند در نتیجه باید سرور را به گونه‌ای پیاده‌سازی کنید که پیام‌ها را با احتمال ۷۰ درصد پاسخ دهد.

پیام‌های پینگ را به فرمت Ping sequence\_number ارسال کنید که sequence\_number شماره پکت ارسال شده (۱ تا ۱۰) است و time زمان فرستاده شدن آن توسط کلاینت است.

در سرور پیام‌های دریافت شده از کلاینت و زمان رسیدن پیام را چاپ کنید. در کلاینت هم علاوه بر چاپ کردن پیام دریافت شده از سرور و زمان پینگ، در صورت عدم دریافت پاسخ از طرف سرور پیام "Request timed out" را چاپ کنید.

به یکی از دو سوال زیر پاسخ دهید: (۱۲۰ نمره)

۲- قبیله‌ای  $n$  نفر عضو دارد (رئیس قبیله جزو افراد قبیله محسوب نمی‌شود) که  $m > 0$  نفر از افراد قبیله به بیماری زردی مبتلا هستند. چون این قبیله از امکانات مدرن مثل آینه بهره‌مند نیست، هیچ کس از زردی خود آگاه نیست ولی هر کس از زردی سایر افراد آگاه است.

رئیس قبیله برای جلوگیری از شیوع بیماری صحبت کردن افراد با یکدیگر را منع کرده است و هر کس فقط با رئیس قبیله صحبت می‌کند. او به همه گفته است در روزی که مطمئن شدید خودتان زردی دارید برای من یک پیام "I\_am\_yellow" بفرستید.

این فرایند را از طریق ساختار سرور (رئیس قبیله) و کلاینت (افراد قبیله) و با پروتکل TCP پیاده‌سازی کنید. ابتدا تعداد  $n$  و  $m$  به عنوان ورودی به سرور داده می‌شود. سرور به اندازه زمان دلخواه  $T$  (برای مثال ۱۰ ثانیه) منتظر می‌ماند تا  $n$  کلاینت به آن متصل شوند و عدد ۱ تا  $n$  را به ترتیب وصل شدن کلاینت‌ها به آن‌ها نسبت می‌دهد. اگر در زمان  $T$  همه  $n$  کلاینت وصل نشوند، سرور خطا می‌دهد و اجرای آن متوقف می‌شود. وگرنه، سرور به صورت تصادفی  $m$  فرد را بیمار می‌کند و به هر کلاینت شماره تمام بیماران به غیر خودش را اعلام می‌کند. حال فرایند تشخیص بیماران شروع می‌شود.

رئیس قبل صبح هر روز برای همه اعضای قبیله پیام "Day <day\_number> started" ارسال می‌کند و تا روزی که همه بیماری خود را اعلام کرده باشند، هر روز این کار را تکرار می‌کند. پس از این یک پیام "END" برای همه افراد قبیله ارسال می‌کند. سرور و همه کلاینت‌ها با دریافت پیام "END" اتصال TCP خود را می‌بندند و برنامه تمام می‌شود. راهنمایی:

- با دیدن پیام شروع روز جدید اعضای قبیله نتیجه می‌گیرند روز قبل کسی بیماری خود را اعلام نکرده است.
- به ازای هر  $m > 0$  دقیقاً در روز  $m$  ام همه  $m$  نفر پیام ذکر شده را به رئیس قبیله اعلام می‌کنند.
- برای تعیین مدت زمان روز می‌توانید از یک زمان‌سنج استفاده کنید.

۳- در این سوال یک ماشین حساب آنلاین پیاده‌سازی می‌کنیم. سرور برنامه نقش ماشین حساب را دارد و با گرفتن دو عدد و یک عملگر از کلاینت عملیات موردنظر را روی اعداد انجام داده و خروجی را برای کلاینت می‌فرستد.

شما باید هر دو برنامه سرور و کلاینت را پیاده‌سازی کنید و برای ارتباط بین آنها از پروتکل TCP استفاده کنید. سرور پس از اجرا شدن منتظر می‌ماند تا کلاینت متصل شود. کلاینت پس از اتصال به سرور با ارسال دستور "start" می‌تواند درخواست انجام محاسبه را به سرور بدهد. سرور پس از دریافت این پیام رشته "OK" را برای کلاینت ارسال می‌کند و منتظر می‌ماند تا کلاینت اعداد را برایش بفرستد. کلاینت پس از دریافت "OK" دو عدد صحیح و یک عملگر از بین عملگرهای  $+$ ،  $-$ ،  $*$ ،  $/$  را هر کدام در یک خط و به ترتیب برای سرور می‌فرستد. سرور با دریافت اعداد و عملگر مورد نظر محاسبات را انجام داده و حاصل را برای کلاینت می‌فرستد و سپس همین چرخه دوباره تکرار می‌شود. در هر مرحله اگر کلاینت رشته "exit" را برای سرور بفرستد سرور اتصال را قطع می‌کند. برنامه کلاینت را باید طوری بنویسید که پس از اتصال به سرور در یک حلقه از کاربر ورودی بگیرد و آن را برای سرور بفرستد و دیتای دریافتی از سرور را نیز چاپ کند. حلقه زمانی به پایان می‌رسد که کاربر عبارت "exit" را وارد کند. همچنین اگر کاربر در هر مرحله دستور غیرمجاز وارد کند سرور باید برای کلاینت یک رشته حاوی متن خطای مناسب را ارسال کند و به ابتدای حلقه باز گردد. (منتظر دستور "start" یا "exit" شود)