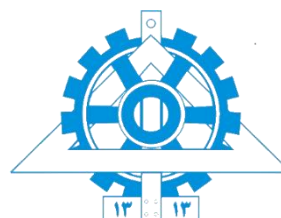


به نام خداوند رنگین کمان



تمرین برنامه نویسی شماره ۰۱



عنوان: پروژه برنامه نویسی - شماره ۰۱

درس: شبکه‌های کامپیوتری

استاد راهنما: دکتر ناصر یزدانی^۱

رشته: مهندسی کامپیوتر

دستیاران آموزشی: اسامه ایران‌دوست^۲، پرمیدا زرغامی^۳

نیمسال دوم سال تحصیلی ۱۴۰۱-۰۲

^۱ نشانی پست الکترونیکی: yazdani@ut.ac.ir

^۲ نشانی پست الکترونیکی: osameh.irandoust@ut.ac.ir

^۳ نشانی پست الکترونیک: parmida.zarghami@ut.ac.ir

فهرست مطالب

عنوان پروژه (Hotel Management System).....	۱
این پروتکل چگونه کار می کند؟.....	۱
مقدمه.....	۱-۱
JSON فایل ها.....	۲-۲
config.json.....	۲-۱-۲
UserInfo.json.....	۲-۲-۲
RoomsInfo.json.....	۲-۳-۲
سرور.....	۳-۳
احراز هویت و مدیریت دسترسی.....	۴-۴
رزرواسیون.....	۵-۵
گزینه اول (View user information).....	۵-۱-۵
گزینه دوم (View all users).....	۵-۲-۵
گزینه سوم (View Rooms Information).....	۵-۳-۵
گزینه چهارم (Booking).....	۵-۴-۵
گزینه پنجم (Canceling).....	۵-۵-۵
گزینه ششم (Pass day).....	۵-۶-۵
گزینه هفتم (Edit information).....	۵-۷-۵
گزینه هشتم (Leaving room).....	۵-۸-۵
گزینه نهم (Rooms).....	۵-۹-۵
گزینه صفر (Logout).....	۵-۱۰-۵
مدیریت خطاها (Error handling).....	۶-۹
مدیریت Log ها.....	۷-۱۰
بونس و بخش های امتیازی (Bonus).....	۸-۱۰
جمع بندی و نکات پایانی.....	۹-۱۱

عنوان پروژه (Hotel Management System)

در این پروژه شما به پیاده سازی یک ورژن ساده شده ی رزرواسیون هتل می پردازید. این برنامه از دو بخش کلاینت (Client) و سرور (Server) تشکیل شده است که سرور وظیفه ی خدمات رسانی را به کلاینت دارد. این پروژه شامل چند فایل Json به عنوان ورودی به سرور خواهد بود. (که همراه تمرین پیوست شده است)

این پروتکل چگونه کار می کند؟

هدف از انجام این تمرین آشنایی با Socket programming است. سوکت به ترکیب یک آدرس ماشین (آی پی یا IP) یا نام هاست (HostName) و یک شماره درگاه (پورت یا Port) گفته می شود. در برقراری ارتباط بین کامپیوترها در یک شبکه، دو چیز بسیار مهم است:

۱- آدرس ماشینی که می خواهیم اطلاعاتی از آن بگیریم یا به آن ارسال کنیم.

۲- برنامه ای از آن ماشین که درخواست اطلاعات کرده؛ و یا اینکه می خواهیم اطلاعاتی از آن برنامه کسب کنیم.

این دو، یعنی آدرس ماشین و شماره برنامه، به وسیله ی سوکت در شبکه مشخص می شوند. سوکت یک ارتباط قابل اطمینان جهت انتقال داده ها بین دو ماشین مجزا می باشد. سوکت، برنامه نویسان را از پیچیدگی های فرآیند برقراری ارتباط بین دو ماشین مانند جزئیات کد کردن بسته ها، فرآیند ارسال داده ها در شبکه، ارسال مجدد بسته های خراب و ... دور ساخته و برنامه نویسان، به راحتی قادر به توسعه ی برنامه های تحت شبکه می باشند.

۱- مقدمه

در این پروژه از شما خواسته شده که یک سامانه رزرواسیون هتل را با استفاده از Socket programming پیاده سازی کنید. پروژه شامل دو نوع کاربر (ادمین و کاربر عادی) می باشد. ادمین دسترسی کامل به تمامی امکانات سامانه را خواهد داشت اما کاربر عادی دسترسی محدودتری خواهد داشت. کاربر عادی به عنوان کلاینت به سرور متصل می شود و در سامانه پس از ورود یا ثبت نام اقدام به رزرو یا لغو و یا مشاهده وضعیت اتاق های هتل می نماید. ادمین نیز می تواند تمام اطلاعات مجاز کاربر (به غیر از پسورد) و اطلاعات اتاق ها را مشاهده و همچنین می تواند رزرو اتاقی را لغو نماید. سامانه بر اساس تاریخ رزرو و تاریخ خروج از اتاق طراحی می شود.

شما باید در این پروژه یک برنامه ی سرور با یک کانال ارتباطی ایجاد کنید. این کانال ارتباطی می تواند روی یک پورت دلخواه که از طریق فایل کانفیگ (config.json) به برنامه ی سرور داده میشود اجرا شود. سپس شما به پیاده سازی یک کلاینت که از طریق این پورت دلخواه به این کانال متصل می شود می پردازید. توجه داشته باشید که سرور وظیفه سرو کردن فایل های کامپیوتری که در آن در حال اجراست را دارد. همین طور محلی که برنامه قرار دارد و اجرا می شود به عنوان دایرکتوری اولیه در نظر گرفته می شود.

۲- JSON فایل‌ها

تمامی اطلاعات سامانه در ۳ فایل JSON کنار پروژه پیوست شده است. این فایل‌ها وظیفه نگهداری اطلاعات موردنیاز برنامه را دارند. در نظر داشته باشید که تمامی این فایل‌ها sample بوده و انتظار می‌رود که شما با توجه به سیستم طراحی شده خودتان این فایل‌ها را تغییر داده و استفاده کنید.

۲-۱- config.json

این فایل شامل پورت موردنیاز برای ارتباط با سرور در کانال ارتباطی است.

```
{
  "hostName": "127.0.0.1",
  "commandChannelPort": 8000
}
```

۲-۲- UsersInfo.json

این فایل شامل اطلاعات کاربران می‌باشد. در این فایل ادمین و کاربر عادی مشخص می‌شود. به عنوان مثال:

```
{
  "id": 0,
  "user": "Admin",
  "password": "admin",
  "admin": "true"
},
{
  "id": 1,
  "user": "Morad",
  "password": "1234",
  "admin": "false",
  "purse": "1000",
  "phoneNumber": "00945692356",
  "address": "Canada, Montral"
},
}
```

همانطور که مشاهده می‌کنید کاربر ادمین صرفاً مشخصات فوق را دارد در صورتی که کاربر عادی دارای شناسه یکتا، اسم کاربر، رمز عبور، موجودی پول کاربر، شماره تلفن و آدرس محل زندگی است. برنامه شما باید حداقل دارای یک کاربر ادمین و ۳ کاربر عادی باشد.

RoomsInfo.json – ۲-۳

این فایل شامل اطلاعات اتاق‌های هتل می‌باشد.

```
"rooms":  
[  
  {  
    "number": "101",  
    "status": 0,  
    "price": 100,  
    "maxCapacity": 2,  
    "capacity": 2,  
    "users":  
    [  
    ]  
  },  
  {  
    "number": "102",  
    "status": 0,  
    "price": 73,  
    "maxCapacity": 3,  
    "capacity": 2,  
    "users":  
    [  
      {  
        "id": 1,  
        "numOfBeds": 1,  
        "reserveDate": "25-02-2023",  
        "checkoutDate": "27-02-2023"  
      }  
    ]  
  },  
]
```

همانطور که مشاهده می‌کنید هر اتاق دارای مشخصاتی از قبیل: شماره اتاق، وضعیت پر یا خالی بودن اتاق (۰ یعنی اتاق خالی و ۱ یعنی اتاق پر)، قیمت اتاق به ازای هر نفر، ظرفیت کل اتاق، ظرفیت خالی است و همچنین اطلاعات کاربرانی که اتاق را رزرو کرده‌اند که شامل شناسه‌ی کاربر، تعداد رزرو، تاریخ رزرو و تاریخ تحویل اتاق است.

۳- سرور

سامانه دارای یک سرور با پورت دریافتی خواهد بود که وظیفه سرور مدیریت کلاینت‌ها است. آدرس و پورت از طریق فایل کانفیگ به سرور داده می‌شود. همچنین اطلاعات کاربران و اطلاعات رزرواسیون نیز در اختیار سرور قرار می‌گیرد.

پس از راه اندازی سرور و بالا آمدن سرور، باید تاریخ شروع سامانه خود را وارد کنید تا در سرور ذخیره شود.

```
Command -> setTime <Date Time>
```

```
Help -> Date time format: 25-02-2023
```

در صورت بروز هرگونه خطا در وارد کردن دستورات سرور ۵۰۳ (در بخش ۶ به اجمال توضیح داده شده است) نمایش داده می‌شود. در صورتی که تاریخ در فرمت نادرستی وارد شود سرور ۴۰۱ به کاربر نمایش داده می‌شود.

۴- احراز هویت و مدیریت دسترسی

این بخش شامل ۳ دستور خواهد بود. برای احراز هویت در سرور باید از فایل UserInfo سرور، اطلاعات کاربران سیستم را خوانده و با استفاده از پسورد آن‌ها را شناسایی و احراز هویت کنند.

```
Command -> signin <username> <password>
```

```
Help -> username: User name, password: user password
```

برای ورود به برنامه طبق دستور فوق و با استفاده از command داده شده کاربر می‌تواند وارد سامانه شود. در صورت بروز خطا در ورود کاربر خطای ۴۳۰ به کاربر نمایش داده می‌شود و در صورت ورود موفقیت آمیز پیغام ۲۳۰ به کاربر نمایش داده می‌شود.

در صورتی که کاربری جدید بخواهد در سامانه ثبت نام کند از دستور زیر استفاده خواهیم کرد.

```
Command -> signup <username>
```

```
Err -> 311: User Signed up. Enter your password, purse, phone and address.
```

```
Command -> <password>
```

```
Command -> <purse>
```

```
Command -> <phone>
```

```
Command -> <address>
```

توجه داشته باشید بعد از ثبت نام کاربر جدید، در صورتی که اسم کاربر وجود داشته باشد خطای ۴۵۱ به کاربر نمایش داده می‌شود. در صورت ثبت نام کاربر و موجود نبودن username پیغام ۳۱۱ نمایش داده می‌شود و بعد از آن کاربر به ترتیب سایر مقادیر خواسته شده را وارد می‌کند. در صورت بروز هرگونه خطایی در وارد کردن مقادیر خطای ۵۰۳ نمایش داده خواهد شد و ثبت نام کاربر نیز لغو می‌شود. در صورت موفقیت آمیز بودن ثبت نام نیز پیغام ۲۳۱ به کاربر نمایش داده خواهد شد.

۵- رزرواسیون

بخش اصلی و هسته برنامه بخش رزرواسیون است. (صفحه اصلی) در این بخش ابتدا اطلاعات موردنیاز به کاربر نمایش داده می‌شود.

```
1. View user information
2. View all users
3. View rooms information
4. Booking
5. Canceling
6. pass day
7. Edit information
8. Leaving room
9. Rooms
0. Logout
Command -> <Choice number>
```

در محیط کنسول (کلاینت) اطلاعات فوق نمایش داده می‌شود و سپس کاربر گزینه‌ی موردنظر خود را انتخاب می‌کند. در صورت بروز هرگونه خطا یا اشتباه وارد کردن گزینه خطای ۵۰۳ به کاربر نمایش داده می‌شود و دوباره این اطلاعات به نمایش در خواهد آمد.

۵-۱- گزینه اول (View user information)

این دستور برای هر دو نوع کاربر قابل دسترسی است. این بخش تمام اطلاعات کاربر را نمایش می‌دهد. (نحوه نمایش اطلاعات و خلاقیت در آن بر عهده دانشجو است) سپس کاربر به صفحه اصلی هدایت می‌شود.

۵-۲- گزینه دوم (View all users)

این دستور صرفاً برای کاربر ادمین قابل دسترسی است و اگر کاربر عادی این دستور را وارد کند با خطای ۴۰۳ مواجه می‌شود و دوباره به صفحه اصلی سامانه بر می‌گردد. در این بخش اطلاعات تمامی کاربران به غیر از رمز عبور به ادمین نمایش داده می‌شود. (نحوه نمایش اطلاعات و خلاقیت در آن بر عهده دانشجو است) سپس کاربر به صفحه اصلی هدایت می‌شود.

۵-۳- گزینه سوم (View Rooms Information)

این دستور برای هر دو نوع کاربر قابل دسترسی است. این بخش تمام اطلاعات تمام اتاق‌ها را اعم از پر یا خالی نمایش می‌دهد. (نحوه نمایش اطلاعات و خلاقیت در آن بر عهده دانشجو است) سپس کاربر به صفحه اصلی هدایت می‌شود.

اگر اتاقی پر یا نصفه پر باشد کاربران حاضر در اتاق را نیز نمایش دهد که این بخش صرفاً باید برای ادمین قابل مشاهده باشد.

۵-۴- گزینه چهارم (Booking)

این دستور برای کاربر عادی استفاده می‌شود. در این بخش کاربر می‌تواند اتاقی با ظرفیت مشخص و تاریخ دقیق شروع اقامت و پایان اقامت رزرو کند. دقت داشته باشید در این پروژه از شما خواسته شده است رزروها را طوری مدیریت کنید که تداخل بین زمان‌ها نباشد.

برای مثال اگر یک اتاق در تاریخ ۲۵ ام پر باشد (ظرفیت آن کامل باشد) دیگر امکان رزرو در ۲۵ ام وجود ندارد اما اگر نیمه خالی باشد بسته به تعداد تخت‌هایی که کاربر درخواست کرده باید بررسی شود که به اندازه‌ی ظرفیت باقی مانده‌ی اتاق می‌باشد یا خیر. همچنین انتظار می‌رود که بعد از رزرو کردن تعداد ظرفیت اتاق و همچنین کیف پول کاربر تغییر کرده باشد.

```
Command -> book <RoomNum> <NumOfBeds> <CheckInDate> <CheckOutDate>
```

```
Help -> RoomNum: Number of room, NumOfBeds: Number of reservation(per person)
```

```
Help -> CheckInDate: Start date of reservation, CheckOutDate: End date of reservation
```

در ابتدا موجودی کاربر با توجه به قیمت اتاق بررسی شده (دقت شود که قیمت هر اتاق به ازای یک نفر است اگر کاربر تعداد بیشتری بخواهد قسمت اتاق در تعداد ضرب می‌شود) و بعد پر بودن اتاق در تاریخ تایین شده بررسی می‌شود.

- در صورت نداشتن موجودی کافی خطای ۱۰۸ نمایش داده شود.
- در صورتی که ساختار دستور نادرست بود خطای ۵۰۳ نمایش داده شود
- در صورتی که اتاقی با شماره مورد نظر یافت نشد خطای ۱۰۱ نمایش داده شود.
- در صورتی که امکان رزرو به دلیل پر بودن اتاق نبود خطای ۱۰۹ نمایش داده شود.

۵-۵- گزینه پنجم (Canceling)

این دستور زمانی بکار می‌رود که کاربر بخواهد رزرو خود را زودتر از Checkout لغو کند در این صورت رزروکاربر پاک شده و نصف پول به کاربر برگردانده می‌شود.

```
Command -> cancel <RoomNum> <Num>
```

```
Help -> RoomNum: Number of room, Num: Number of reservation(per person)
```

بعد از وارد کردن شماره پنجم می‌بایست لیستی از رزروهای کاربر نمایش داده شود که طریقه‌ی نمایش آن بر عهده دانشجو است. سپس با استفاده از دستور فوق شماره اتاق m به تعداد n نفر را کنسل کنید.

چنانچه اتاق مورد نظر یافت نشد خطای شماره ۱۰۱ و چنانچه رزروی با کاربر مورد نظر با آن تعداد یا بیشتر در اتاق مورد نظر موجود نبود خطای شماره ۱۰۲ را نمایش دهید و اگر شکل دستور نادرست بود خطای ۴۰۱ نمایش داده می‌شود.

در صورت موفقیت امیز بودن دستور ۱۱۰ نمایش داده شود.

دقت شود که سیستم باید زمانی که تقویم سامانه برابر با Checkout هر رزروی شد آن رزرو را پاک کند.

۵-۶- گزینه ششم (Pass day)

این دستور صرفاً برای کاربر ادمین قابل دسترسی است و اگر کاربر عادی این دستور را وارد کند با خطای ۴۰۳ مواجه می‌شود و دوباره به صفحه اصلی سامانه بر می‌گردد. در این بخش ادمین می‌تواند تعداد روزهایی که گذشته است را وارد کند. به عنوان مثال تاریخ وارد شده در سرور ۱۰ اسفند می‌باشد و ادمین با وارد کردن دستور زیر و مقدار ۲ می‌تواند تاریخ را به ۱۲ اسفند تغییر دهد.

```
Command -> passDay <value>
```

۵-۷- گزینه هفتم (Edit information)

این دستور برای هر دو نوع کاربر قابل دسترسی است. در این بخش کاربر می‌تواند اطلاعات وارد شده خود را تغییر دهد. در نظر داشته باشید ادمین نیز مانند سایر کاربران تنها اطلاعات شخصی خود که فقط رمز عبور می‌باشد را می‌تواند تغییر دهد و به اطلاعات سایرین نمی‌تواند دسترسی داشته باشد. (نحوه نمایش اطلاعات و خلایق در آن بر عهده دانشجو است) سپس کاربر به صفحه اصلی هدایت می‌شود. در صورت بروز هرگونه خطا یا اشتباه وارد کردن اطلاعات خطای ۵۰۳ به کاربر نمایش داده می‌شود و به صفحه اصلی هدایت می‌شود و در صورت موفقیت آمیز بودن پیام ۳۱۲ نمایش داده می‌شود.

```
Command -> <new password>
```

```
Command -> <phone>
```

```
Command -> <address>
```

۵-۸- گزینه هشتم (Leaving room)

این دستور برای هر دو نوع کاربر قابل دسترسی است. کاربر عادی می‌تواند پیش از موعد اتاق را تحویل دهد و از اتاق خارج شود.

```
Command -> room <value>
```

```
Help -> value: Room number
```

در صورتی که کاربر در اتاق سکونت داشته باشد پیام ۴۱۳ و در صورتی که اتاق اشتباهی وارد شود خطای ۵۰۳ و در صورتی که اتاقی که کاربر در آن سکونت نداشته باشد وارد گردد نیز خطای ۱۰۲ به کاربر نمایش داده می‌شود. تغییرات نیز بعد از خروج کاربر ذخیره می‌شود.

(امتیازی) در این بخش ادمین می‌تواند هر اتاقی که بخواهد را خالی کند و Status اتاق را تغییر دهد. در مواردی که فردی که اتاقی را رزرو می‌کند در صورت تخطی از قوانین هتل مدیریت مجبور به حذف رزرو و خالی کردن اتاق می‌شود.

```
Command -> room <value>
```

```
Command -> capacity <new capacity>
```

```
Help -> value: Room number
```

```
Help -> new capacity: Enter new capacity of the room
```

در صورتی که شماره اتاق اشتباه وارد شود خطای ۱۰۱ و در صورتی که ظرفیت وارد شده توسط کاربر بیشتر از ظرفیت مجاز اتاق باشد خطای ۴۱۲ به کاربر نمایش داده می‌شود و در صورت موفقیت آمیز بودن کار، بعد از ذخیره تغییرات پیام ۴۱۳ به کاربر نمایش داده می‌شود.

۹-۵- گزینه نهم (Rooms)

این دستور صرفاً برای کاربر ادمین قابل دسترسی است و اگر کاربر عادی این دستور را وارد کند با خطای ۴۰۳ مواجه می‌شود و دوباره به صفحه اصلی سامانه بر می‌گردد. در این صفحه ادمین می‌تواند با استفاده از سه دستور اتاقی را اضافه کند یا تغییر دهد و یا حذف کند.

دستوارت به صورت زیر می‌باشند:

```
Command -> add <RoomNum> <Max Capacity> <Price>
Command -> modify <RoomNum> <new Max Capacity> <new Price>
Command -> remove <RoomNum>
```

- در صورت وجود نداشتن شماره‌ی اتاق خطای ۱۰۱ را نمایش دهید
 - در صورت وجود داشتن شماره اتاق در دستور Add خطای ۱۱۱ را نمایش دهید.
 - در صورت موفقیت در اضافه کردن اتاق پیام ۱۰۴ و در تغییر اتاق پیام ۱۰۵ و پاک کردن اتاق پیام ۱۰۶ را نمایش دهید.
- دقت داشته باشید که در صورت پر بودن اتاق امکان حذف اتاق و همینطور کم کردن ظرفیت اتاق وجود نداشته و برای ادمین باید خطای 109 نمایش داده شود.

۱۰-۵- گزینه صفر (Logout)

این دستور برای هر دو نوع کاربر قابل دسترسی است. با زدن این دستور کاربر از برنامه خارج می‌شود و ارتباط همان کلاینت با سرور قطع می‌شود. در صورت موفقیت آمیز بودن خروج کاربر پیغام ۲۰۱ نمایش داده خواهد شد.

نکته ۱: در تمامی قسمت‌ها انتظار می‌رود بعد از چاپ پیام مناسب هر دستور کاربر به صفحه اصلی باز گردد.

نکته ۲: بعد از پایان برنامه باید تمامی فایل‌های Json به‌روزرسانی شوند.

۶- مدیریت خطاها (Error handling)

در این بخش کد خطاها و پیغام‌ها با ذکر جزئیات توضیح داده شده است.

```
Err -> 101: The desired room was not found
Err -> 102: Your reservation was not found
Err -> 104: Successfully added.
Err -> 105: Successfully modified.
Err -> 106: Successfully deleted.
Err -> 108: Your account balance is not enough
Err -> 109: The room capacity is full
Err -> 110: Successfully done.
Err -> 111: This room already exists
Err -> 201: User logged out successfully.
Err -> 230: User logged in.
Err -> 231: User successfully signed up.
Err -> 311: User Signed up. Enter your password, purse, phone and address.
Err -> 312: Information was changed successfully.
Err -> 401: Invalid value!
Err -> 403: Access denied!
Err -> 412: Invalid capacity value!
Err -> 413: successfully Leaving.
Err -> 430: Invalid username or password.
Err -> 451: User already existed!
Err -> 503: Bad sequence of commands.
```

۷- مدیریت Log ها

سرور شما در زمان اجرا باید یک فایل log در کنار خود ایجاد کند و تمامی اطلاعات را با تاریخ و ساعت وقوع در آن ذخیره کند. اطلاعاتی مانند افرادی که وارد سیستم شده‌اند؛ رزروهایی که انجام شده و تمامی مواردی که در بخش ۵ وجود دارد. هر کاربری که دسترسی دارد می‌کند باید لاگ آن نیز ذخیره شود. همچنین به ازای هر کاربر یک فولدر در Client برنامه هم اسم کاربر ساخته می‌شود و لاگ‌های مربوط به هر کاربر در فولدر مربوطه ذخیره می‌شود.

دقت کنید با هر بار اجرای سرور اگر فایل log وجود نداشت آن را بسازید و در صورتی که از قبل این فایل وجود داشته باشد، در ادامه‌ی آن شروع به نوشتن کنید.

۸- بونس و بخش‌های امتیازی (Bonus)

بونس‌های این تمرین حداکثر ۱۰ نمره (از ۱۰۰) خواهد بود.

- فیلتر اتاق‌ها (Room filtering)

در بخش ۳-۵ شما می‌توانید با افزودن دو گزینه دیگر به کاربر این امکان را بدهید که صرفاً اتاق‌های خالی را ببیند یا بتواند تمامی اتاق‌ها را مشاهده کند. (۵ نمره)

- بخش ۸-۵ نیز قسمت ادمن امتیازی است. نحوه نمایش آن نیز در همان بخش توضیح داده شده است. (۵ نمره)

۹- جمع بندی و نکات پایانی

- مهلت تحویل: ۲۱ اسفند
- پروژه در گروه‌های ۲ نفره انجام می‌شود. (گروه بندی در سامانه ایلرن نیز انجام می‌شود و تحویل تمرین به صورت گروهی خواهد بود)
- برنامه شما باید درخواست‌ها را به صورت همزمان پاسخ دهد و چندین کلاینت بتوانند با کاربران مختلف به سرور وصل شوند. (می‌توانید از thread یا system call select() به دلخواه خودتان استفاده کنید. در ضمن توجه داشته باشید که ۲ سرور بر روی یک پورت به صورت همزمان بالا نیاید!)
- هر ۲ نفر می‌بایست کار را تقسیم کنند. همچنین از Git برای ساختن branch و تقسیم issue ها استفاده نمایید. (با استفاده از commit ها و تعیین issue ها میزان مشارکت هر نفر مشخص می‌شود). بعد از انجام این کار کدها را در یک repository به نام CN_CHomeworks_1 در اکانت‌های GitHub/GitLab خود قرار دهید (به صورت private). همچنین در یک فایل README.md می‌توانید report و داکيومنت خود را کامل کنید و در کنار repository قرار دهید. در نهایت لینک این repository را در محل پاسخ تمرین قرار دهید. (از فرستادن فایل به صورت زیپ جدا خودداری نمایید). اکانت تی ای‌های این تمرین رو به Repo خودتون به عنوان Maintainer به پروژه اضافه کنید.

Git Accounts:

@osameh15

@pzarghami

- برای پیاده سازی این تمرین از C یا C++ استفاده کنید. (می‌توانید از محیط توسعه QT و تمامی کتابخانه‌های آن نیز استفاده کنید)
- سیستم عامل ترجیحا Linux باشد.
- دقت کنید گزارش نهایی شما می‌بایست همانند یک Document باشد و شامل توضیح کد و ساختار کد، همچنین نتیجه نهایی اجرای کد و اسکرین شات‌های دقیق از تمام مراحل باشد. (در فایل Readme.md کنار فایل اصلی خود و در Repo مربوطه قرار دهید). **این نکته حائز اهمیت است که فایل PDF به هیچ عنوان مورد پذیرش قرار نخواهد گرفت.**
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- برای هر قسمت کد، گزارش دقیق و شفاف بنویسید. کدهای ضمیمه شده بدون گزارش مربوطه نمره‌ای نخواهند داشت.
- هدف این تمرین یادگیری شماست. لطفا تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، مطابقت سیاست درس با گروه متقلب و تقلب دهنده برخورد خواهد شد.
- سؤالات خود را تا حد ممکن در **فروم درس** مطرح کنید تا سایر دانشجویان نیز از پاسخ آن بهره‌مند شوند. در صورتی که قصد مطرح کردن سؤال خاص‌تری دارید، از طریق ایمیل زیر ارتباط برقرار کنید. **توجه داشته باشید** که سایر شبکه‌های اجتماعی راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.

Osameh.irandoust@ut.ac.ir ○

Parmida.zarghami@ut.ac.ir ○

موفق باشید