دانشگاه صنعتی شریف

دانشکده مهندسی برق

آزمایشگاه برنامه نویسی پایتون

" Python - Git "

Git

1-1: در این آزمایشگاه، از GitHub برای مدیریت پروژه و ارائه آن به دستیاران آموزشی استفاده می‌شود. بدین منظور در این سایت یک اکانت ساخته و سپس هر تیم یک Repository خصوصی با عنوان Python-Lab و شماره گروه، ساخته و آن را با بقیه اعضای تیم و دستیار آموزشی گروه، به اشتراک بگذارید.

1-2: یکی از اعضای تیم، در سیستم Local خود، یک فایل با نام (Lab3) ساخته و با تنظیم درست Config، آن را در مخزن مربوطه آپلود کند. بقیه اعضا این فایل را دریافت کنند و بنابر نحوه تقسیم تمارین در گروه، کدهای خود را در آن بارگزاری کنند. در خصوص remote در گیت مطالعه کرده و عملیات آن را شرح دهید.

1-3: بعد از انجام تمارین بخش پایتون، هر یک اعضا کد خود را جهت بررسی به گیت مربوطه در قالب Merge Requests ارسال کنید.

در سوالات، هرجایی که ذکر شده است: **"شرح دهید."** در توضیحات Merge Requests در حد ۲ الی ۳ خط در خصوص آن مورد توضیح دهید. سپس بررسی Merge Requests را به دستیار آموزشی محول کنید.

امتیازی: در خصوص نحوه Assign کردن بررسی کد و همچنین نحوه دریافت و ارائه Task در پلتفرم GitLab توضیح دهید. با مثال عملی این روند را شرح دهید.

1-4: در صورتی که دستیار آموزشی کد درخواستی جهت Merge Requests را تائید کرد، عملیات Merge کردن کد هر یک از اعضا بر Master Branch را طی کنید. معیار نمره نهایی، کد و Commit های Master Branch می‌باشد.

1-5: در خصوص Diff, Remote, Reset, Rebase مطالعه کرده و کارکرد هرکدام را به صورت مختصر شرح دهید.

امتیازی: در مورد Cherry pick مطالعه کرده و قابلیت‌های آن را شرح دهید.

Fundamentals – Python

2-1: یک لیست ایجاد کرده و عملکرد دستور append را با اضافه کردن شماره دانشجویی و نام خود به انتهای لیست بررسی نمایید. چند لیست دیگر به همین شکل ایجاد نمایید و سپس تمام لیست‌های ایجاد شده را در یک لیست قرار دهید. نحوه‌ی اندیس‌دهی برای دسترسی به اعضای لیست که شامل لیست‌های درونی و متغیرهای آنها هست را شرح دهید.

2-2: بخش قبل را این بار با استفاده از Tuple مجددا انجام دهید. به تفاوت‌های این قسمت با قسمت قبل، اشاره کنید.

2-3: دو لیست به صورت زیر ایجاد کنید، سپس با استفاده از دستور zip یک دیکشنری در قالب زوج مرتب با این دو لیست ایجاد کنید. حال با دستور dict دیکشنری ایجاد شده را به یک دیکشنری با فرمت Key – Value تبدیل نمایید. نحوه دسترسی به مقادیر از طریق Key ها و Valueها را شرح دهید.

```
nation = ['Roman','Egypt','Greek','Chinese','Islamic','Mayan','Persian','Mongol']

golden_age = ['27BC-1453AD','3150BC-30BC','800BC-600AD','221BC-1912AD','750AD-1257AD','2000BC-1540AD','550BC-651AD','1206AD-1368AD']
```

2-4: قابلیت‌های list comprehension در پایتون را مطالعه کرده و عملکرد هر یک از خطوط کد زیر را شرح دهید.

```
a = [2**i for i in range (17)]

b = [x*x for x in range (1,5)]

c = [a[x] for x in b if x % 2 == 0]
```

2-5: لیستی از لیست‌ها ایجاد کنید به صورتی که هر عضو آن، دارای دو عضو درونی از جنس رشته (مثلا نام و نام خانوادگی چند نفر) است. برنامه‌ای بنویسید که با استفاده از روش list comprehension[1] و توابع Lambda، در یک خط لیست مذکور را تبدیل به لیستی نمایید که هر عضو آن نام و نام خانوادگی شخص به صورت یک رشته باشد.

[1] https://en.wikipedia.org/wiki/List_comprehension

Python In Use

3-1: virtual environmentها می‌توانند جهت ایجاد فضای مستقل برای نصب و استفاده از نسخه‌های متفاوت پایتون به کار روند. با استفاده از دستور مناسب، یک environment با نام خودتان ایجاد کنید و نحوه‌ی فعال کردن و غیرفعال کردن این environment و مشاهده‌ی لیست environment های موجود را شرح دهید[۲].

3-2: فایلی با نام خود و پسوند py. ایجاد کنید و در آن تابعی بنویسید که در ورودی یک عدد از کاربر دریافت کند و مجذور آن را برگرداند. در فایل اصلی خود، ماژولی که ساخته اید را import کنید و با دریافت عدد ورودی از کاربر و استفاده از تابعی که نوشتید، مجذور عدد مورد نظر را محاسبه و چاپ نماید. نحوه‌ی کارکرد from...import را شرح دهید.

3-3: برنامه ای بنویسید که ابتدا یک فایل به نام 'python_lab.txt' ایجاد کرده و در حالت نوشتاری آن را باز کند. سپس نام خود را در فایل بنویسید و فایل را ببندید. در ادامه مجددا فایل را با دستور with باز کرده و بدون از بین بردن اطلاعات موجود در فایل، شماره ی دانشجویی‌تان را در خط بعد فایل بنویسد. عملکرد دستورات read و readlines و with را برای خواندن از فایل شرح دهید.

3-4: برنامه ای بنویسید که ابتدا بررسی کند آیا دایرکتوری 'input_files' در محل فایل اصلی برنامه وجود دارد یا خیر و در صورتی که وجود ندارد این دایرکتوری را ایجاد کند. سپس بدون قرار دادن hardcode آدرس کامل اجرای فایل اصلی، مشابه تمرین قبل فایلی در دایرکتوری input_files ایجاد کنید و نام خود را در آن بنویسید.

3-5: فایل 'data.csv' که در اختیارتان قرار داده شده است را بخوانید. سپس داده ابتدای هر سطر که شماره‌ی سطر است را حذف نمایید و نتیجه را با کاراکتر 'whitespace' در فایل جدیدی با همین فرمت بنویسید.

3-6: ماژول re[3] برای استفاده از regular expression در زبان پایتون تعبیه شده است. تمرین بخش 3-5 از آزمایشگاه 1 را این بار با استفاده از توابع match، search، find و sub این ماژول انجام دهید.

---

[2] https://conda.io/docs/using/envs.html

[3] https://docs.python.org/3/howto/regex.html

Algorithm – Python

4-1: برنامه ای بنویسید که با استفاده از روش بازگشتی دو رشته از کاربر دریافت کند و فاصله‌ی Levenshtein بین دو رشته را بدست آورد.

امتیازی: سوال را با روش برنامه‌نویسی پویا پیاده‌سازی کنید.

4-2: برنامه ای بنویسید که یک عدد از کاربر دریافت کند و دنباله ی Collatz[4] را با شروع از این عدد تولید نموده و نمایش دهد. سپس برنامه را به نحوی ویرایش کنید که در بین اعداد آغازی زیر 1000، عددی را بیابد که دنباله ی Collatz آن بزرگترین طول ممکن را داشته باشد و عدد مربوطه و دنباله ی آن را چاپ کند.

4-3: جدولی باینری به صورت زیر در نظر بگیرید. برنامه ای بنویسید که مرکز ثقل جدول را بیابد و اندیس مربوط به آن را چاپ نماید. مرکز ثقل جدول نقطه ای است که اگر قرار باشد از تک تک خانه هایی که شماره ی ۱ دارند به آن خانه برویم، مجموع فواصل طی شده حداقل مقدار ممکن باشد. (فقط حرکات افقی و عمودی مجاز است)

```
Table = [[1, 0, 0, 0, 1],
         [0, 0, 0, 0, 0],
         [0, 0, 1, 0, 0]]
```

```
C = [0,2]
Distance = 2 + 2 + 2 = 6
```

4-4: جدولی مربعی از حروف به صورت زیر در نظر بگیرید. برنامه ای بنویسید که یک رشته از کاربر دریافت کند و بررسی کند که آیا این رشته توسط تعدادی از حروف مجاور جدول قابل تولید شدن هست یا خیر.

```
Table = [["ABCE"],
         ["SFCS"],
         ["ADEE"]]
```

```
word = "ABCCED"   -> returns true,
word = "SEE"      -> returns true,
word = "ABCB"     -> returns false.
```

امتیازی: فرض کنید با خودرو قصد سفر بین دو شهر را دارید. خودروی شما گنجایش C لیتر بنزین را دارد و مصرف سوخت آن F لیتر در کیلومتر است. در طول مسیر تعداد N پمپ بنزین وجود دارد که در فاصله‌ی D[i](i=1toN) کیلومتری از مبدا قرار دارند و نرخ سوخت در هر پمپ بنزین P[i](i=1toN) تومان برای هر لیتر است.

با این فرض که در ابتدا با باک پر حرکت کرده اید و در هر پمپ بنزین نیز یا توقف نمی‌کنید، یا اگر توقف کنید باک را پر می کنید، برنامه ای بنویسید که با داشتن C، F، N، D، و P به شکل بازگشتی هزینه کمینه را محاسبه نماید.

---

[4] https://en.wikipedia.org/wiki/Collatz_conjecture