

تمرین های سری اول درس Service Based Architecture

۱. برنامه ای بنویسید که سه نمره ی دانشجو را از کاربر بگیرد.

اعداد صحیح یا اعشاری می توانند باشند. حاصل جمع، حاصل ضرب و میانگین اعداد را چاپ کند. به این صورت:

Please enter 3 integers: 10 13 12.5

Sum is: 35.5

Product is: 16.25

Average is: 11.833

همچنین در صورتی که میانگین نمره ها از 18 بیشتر باشد، در خروجی مقدار A، میانگین بین 16 تا 18 بود مقدار B و در غیر اینصورت مقدار C را در خروجی چاپ نماید.

۲. (آ) یک اسکریپت پایتون با دیکشنری بنویسید که کلیدها اعداد بین ۱ تا ۱۵ باشند و مقادیر مربع کلیدها هستند (1 : 1, 2 : 4, 3 : 9, ...).

(ب) از کاربر نام و معدل ۱۰ دانشجو را بگیرید و در دیکشنری (کلید : نام، مقدار : معدل) قرار دهید.

۳. برنامه ای بنویسید که یک کلمه یا جمله از کاربر بگیرد و در صورتی که این کلمه با مقدار عکس خود یکی باشد، مقدار true و در غیر این صورت مقدار false را چاپ نماید (برای مثال hi ih مقدار true و tree مقدار false است.)

۴. برنامه ای بنویسید که از کاربر عدد بگیرد تا زمانی که کاربر عدد صفر را وارد کند. بعد به روش bubble sort این اعداد را مرتب کنید و در خروجی چاپ کنید.
راهنمایی:

روش bubble sort یک الگوریتم مرتب سازی ساده است که فهرست را پشت سرهم پیمایش می کند تا هر بار عناصر کناره را با هم سنجیده و اگر در جای نادرست بودند جابه جایشان کند. در این الگوریتم این کار باید تا زمانی که هیچ جابه جایی در فهرست رخ ندهد، ادامه یابد و در آن زمان فهرست مرتب شده است.

۵. در شاخه ای که فایل پایتون شما وجود دارد یک فلدر به نام root بسازید. هدف این است که داخل این فلدر تعدادی فایل و فلدر دیگر بسازیم به این صورت:

```

۱ root: | .txt
۲       | .txt
۳       | .txt
۴       | First: | .txt
۵           | .txt
۶           | .txt
۷           | Second: | .txt
۸               | .txt
۹               | .txt
۱۰              | Third: | .txt
۱۱                  | .txt
۱۲                  | .txt
۱۳                  | Fourth: | .txt
۱۴                      | .txt
۱۵                      | .txt
۱۶                      | Fifth: | .txt
۱۷                          | .txt
۱۸                          | .txt

```

این فلدر تا ۵ مرحله شاخه بندی شده است. داخل هر فلدر تعدادی فایل متنی وجود دارد که با .txt داخل شکل نشان داده شده اند. اسم هر کدام از این فایل ها یک عدد تصادفی بین ۰ تا ۱۰۰ می باشد. توجه کنید که تعداد این فایل ها نیز تصادفی می باشد.

(آ) برنامه مورد نظر را طوری بنویسید که با هر بار اجرا کردن فلدر ها و فایل های قبلی پاک شوند (در صورت وجود) و دوباره ایجاد شوند.

(ب) در قسمت دوم همین برنامه با استفاده از تابع `os.walk` ابتدا تعداد کل فایل های ایجاد شده را بشمارید و سپس تمامی فایل هایی که داخل نام فایل عدد صفر را دارند به جای عدد صفر از کاراکتر `'_'` استفاده کنید (`rename`).

راهنمایی: در این تمرین نیاز به استفاده از تعدادی از توابع داخل کتابخانه `os` از جمله `os.path.exists`، `mkdir` و `rename` دارید. همچنین از توابع کتابخانه `shutil` مثل `rmtree` نیز می توانید استفاده کنید. با توابع رشته ای هم می توانید جهت عوض کردن اسم فایل کار کنید.

۶. (آ) **امتیازی-** یک فایل متنی درست کنید به این ترتیب که متشکل از 1 000 000 خط باشد که از شماره خط صفر شروع می شود. هر خط این فایل به این شیوه نوشته شده باشد:

```

۱ Line 000000      RandomNumber
۲ Line 000001      RandomNumber
۳ .

```

۴ .
۵ .

برای تولید عدد تصادفی بین صفر و یک از تابع `random` داخل کتابخانه `numpy.random` استفاده کنید و با ۶ رقم بعد از ممیز عدد را نشان دهید. توجه کنید که در این فایل بعد از کلمه `Line` از فاصله استفاده شده و بعد از شماره خط از `TAB` استفاده می شود. زمان اجرای کار را ذخیره کنید و نشان دهید.

(ب) **امتیازی-** سپس همین کار را دقیقاً توسط کتابخانه `csv` انجام دهید و زمان را اندازه بگیرید و درصد کاهش و یا افزایش را بیان کنید. به نظر شما علت این کاهش یا افزایش چه چیزی می تواند باشد؟ (همه جا از فاصله عادی یا `TAB`) استفاده کنید.

(ج) **امتیازی-** در قسمت سوم این سوال اگر کل داده ها را داخل یک لیست در ابتدا ذخیره کنید و سپس داخل فایل بریزید آیا سریع تر خواهد بود؟ در صورت مثبت و منفی بودن جواب علت را توضیح دهید؟ در کل یک فایل پایتون خواهیم داشت که از سه قسمت تشکیل شده است و در هر قسمت زمان اجرا ذخیره می شود و نشان داده می شود.

راهنمایی: برای قسمت دوم سوال بایستی از تابع `csv.writer` استفاده کنید که ورودی آن `File` `Handle` می باشد و خروجی آن `CSV Handle` خواهد بود. متغیر اخیر یک متد به نام `writerow` دارد که یک لیست را ورودی می گیرد و در خروجی چاپ می کند. در قسمت سوم سوال از `List Comprehension` استفاده کنید. کل عملیات باز کردن فایل و ریختن داده ها در دو خط کد قابل انجام خواهد بود. از تابع `writelines` استفاده بفرمایید.