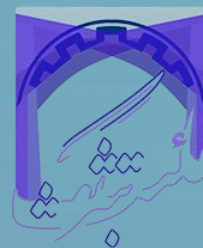


مکتب شریف

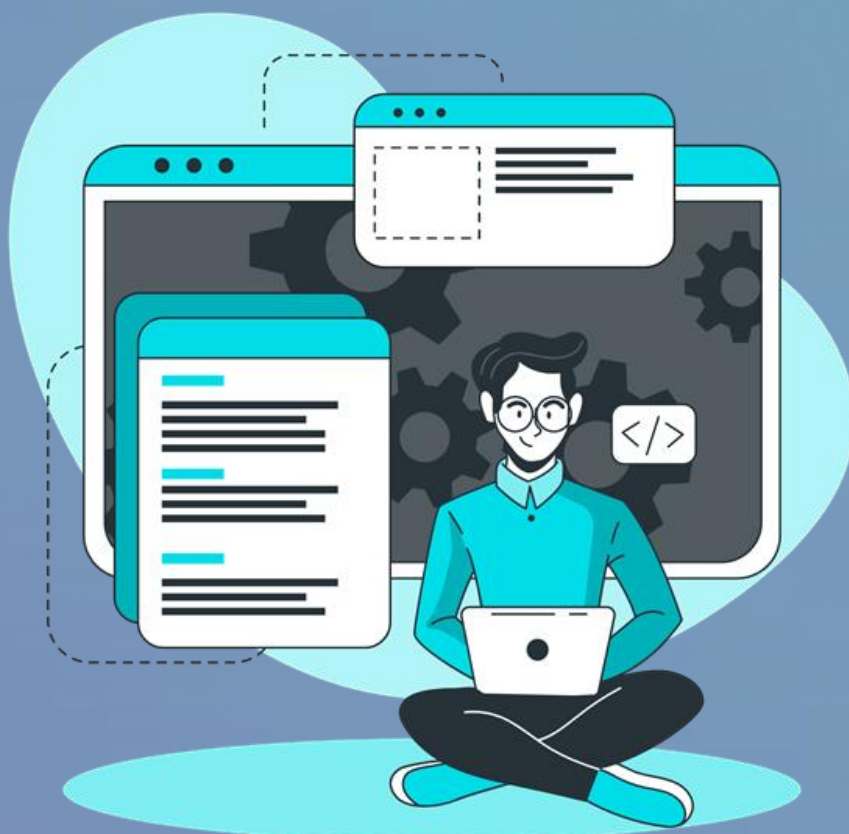
اولین بوتکمپ آموزشی - استخدامی ایران



PYTHON

BOOTCAMP

PRACTICE #03



۱- در دیجی کالا برای ذخیره سازی برخی از رشته های عددی از نوعی فشرده سازی استفاده میشود تا کسی نتواند از خروجی تولید شده به رشته ی عددی اصلی دست پیدا کند.

۱.۱- البته ما در اینجا از این نکته که این روش تصادم دارد؛ به این معنا که چند ورودی مختلف ممکن است خروجی یکسانی تولید کنند، چشمپوشی میکنیم! لازم به ذکر است که رشته ی عددی فقط شامل ارقام ۰ تا ۹ است.

۲.۱- الگوریتم به این صورت است که تعداد تکرار همه ی ارقام رشته را حساب میکند، ارقام تکراری را حذف میکند و تعداد تکرار هر رقم (با شرط این که حداقل ۲ بار تکرار شده باشد) را در رشته ورودی مینویسد و در نهایت رشته ی عددی را به صورت صعودی مرتب میکند .

۳.۱- این کار روی خروجی به دست آمده مجدد تکرار می شود و آنقدر ادامه دارد تا خروجی نهایی با خروجی مرحله ی قبل تفاوتی نکند.

تذکره:تابع عملیات بالا باید پیاده سازی و ایجاد شود .

ورودی نمونه و خروجی نمونه :

تنها عبارت ورودی رشته عددی مورد نظر است که میخواهیم آن را فشرده کنیم.

طول این رشته حداقل یک و حداکثر 1000 رقم است.

تنها عبارت خروجی، عبارت فشرده شده نهایی است .

```
...
enter digit string : 442254545
22345
>>>
```

توضیح:

در این مثال رقم 4، 4 بار تکرار شده است و رقمهای 2 و 5 هم به ترتیب 2 و 3 بار تکرار شده اند. رقم های تکراری حذف میشوند و فقط یکی از آنها در رشته باقی میماند، بنابراین رشته ی 425 باقی میماند. سپس تعداد تکرار هر رقم در ادامه ی رشته نوشته میشود، بنابراین رشته ی 425423 ایجاد میشود و در نهایت ارقام به صورت صعودی مرتب میشوند که در این جا 223445 حاصل میشود.

مجدداً عملیات فشرده سازی روی این رشته ی حاصل شده اعمال میشود و نتیجه ی آن 222345 میشود. یک بار دیگر عملیات فشرده سازی اعمال میشود و نتیجه ی آن 23345 میشود و با اعمال مجدد این الگوریتم خروجی 22345 حاصل میشود که دیگر قابل فشرده سازی نیست.

۲- یک برنامه بنویسید که سبد خرید یک فروشگاه آنلاین ساده را شبیه سازی کند که شامل افزودن محصول ، حذف محصول ؛ محاسبه ی قیمت کل سبد خرید یک مشتری و همینطور نمایش سبد خرید مشتری باشد .

- از نام مشتری برای ایجاد سبد خرید مربوط به مشتری استفاده کنید . به طوری که اگر دیکشنری مربوط به مشتری وجود داشت آن را به روز کرده و در غیر این صورت آن را ایجاد کند

- لازم به ذکر است که در مواقع خواندن سبد خرید مشتری ، در صورت عدم وجود این سبد خرید از exception handling مناسب استفاده کنید

از انجایی که هیچ ایده برای ذخیره سازی دیتابیس ندارید . از حلقه بینهایت استفاده کنید که پس از ساخت دیتا برنامه متوقف نشود

```
>>> add_item(Hossein , 'laptop' , 1300 , 2)
2 laptop added to Hossein Cart lists thats equal to : 2600$
>>>
```

- برای توابع نام برده شده از فرمتی شبیه به بالا با تعداد ورودی های کمتر یا بیشتر استفاده کنید . که در مثال بالا ، ورودی اول نام مشتری که برابر است با نام دیکشنری سبد خرید ، ورودی دوم ؛ نام محصول ، ورودی سوم قیمت و ورودی چهارم تعداد محصول بوده که میتواند مقدار پیشفرض داشته باشد .

۳- ابتدا تابعی برای برگرداندن تمامی کلمات یکتای یک جمله بنویسید ؛ سپس بدون استفاده از حلقه ها، این تابع را روی لیستی از اعداد اعمال کرده و لیستی جدید تشکیل دهید

```
>>>
>>> input_sentences = [
...     "This is a sample sentence.",
...     "Python programming is fun."
... ]
>>>
>>> unique_sorted_words = get_unique_sorted_words(input_sentences)
>>> print(unique_sorted_words)
['Python', 'This', 'a', 'fun.', 'is', 'programming', 'sample', 'sentence.']
>>>
```

۴- برنامه ای بنویسید که یک پاراگراف در ورودی گرفته و تعداد کلماتی که بیشتر از همه تکرار شده اند را به همراه تعداد آنها برگرداند .

```
>>>
>>> input_paragraph = "This is a sample paragraph. It contains several words, some of which are repeated. This is a good exercise to find the most frequent words."
...
>>> result_words, result_frequency = find_most_frequent_words(input_paragraph)
...
>>> print("Most Frequent Word(s):", result_words)
Most Frequent Word(s): ['this', 'is', 'a']
>>> print("Frequency:", result_frequency)
Frequency: 2
>>>
```

نکات

- مهلت ارسال تمرین تا **پایان ساعت ۹ صبح روز پنجشنبه** می باشد.
- نام فایل ارسالی خود را به این صورت قرار دهید. به عنوان مثال.
 ○ **Firstname_Lastname_HWNumber_maktabNumber**
 ○ **Mohammad_Samadi_HW03_maktab112**
 ○ ⚠️ دقت فرمایید که منظور از **Number** - بعد از **HW** شماره تمرین می باشد.
 ○ ⚠️ دقت فرمایید که منظور از **Number** - بعد از **Maktab** شماره بوتکمپ می باشد.
- در مورد تمرینهای پایتون هر تمرین را در یک فایل جداگانه پایتون با پسوند **py** بنویسید و از ارسال فایل **Jupyter** یا مشابه آن خودداری کنید.
- در صورتی که تمرین شامل چند فایل و فولدر می باشد حتماً آنها را در قالب یک فایل فشرده شده تجمیع کنید (**ZIP/ RAR**)
- در صورت لزوم یک فایل **Word** به عنوان توضیح در کنار کدهای خود قرار دهید.
- در صورتی که سوالی دارید در کارتابل گروهی خود از مربیان بپرسید.
- **توصیه دوستانه.** از مواجهه با هیچ سوالی نترسید. به هر میزانی که در حل سوالات پیشروی کرده باشید نمره بخش مورد نظر را دریافت می کنید. بنابراین بیش از آنکه رسیدن به خروجی نهایی مهم باشد، تلاش شما ارزشمندتر است.
- قطعاً هدف از تمارین صرفاً رسیدن به جواب نهایی نیست و تمیز بودن کد و خلاقیتی که در انجام آن به خرج می دهید از اهمیت و امتیاز بالایی برخوردار است. ارائه راه حل کلی و عمومی برای یک مسئله که حالت های مختلف آن را در نظر بگیرد و فراتر از خواسته ی مسئله است. (خواسته ی مسئله گسترش داده شود یا حالت های خاص مسئله را پوشش دهد. قطعاً مشمول امتیاز بیشتری خواهد شد).
- سوالات امتیازی شامل مواردی است که نیازمند سرچ بیشتر شما عزیزان می باشد. بنابراین حل این سوالات نمره امتیازی دارد.

موفق باشید