

سلام و درود 😊 الیاس نجفی هستم. بریم سراغ توضیحات و چالش‌ها...

اولین بار که قسمت اول سوال رو خوندم گفتم خب دیگه خیلی آسونه! بعد که رفتم جلوتر با اولین چالش رو به رو شدم. که پوشه بندی فایل‌ها بود. با اینکه می‌گن این ساختار، خوانش کد رو راحت‌تر می‌کنه، نمیدونم چرا من بیشتر قاطی می‌کنم 😁 و نوشتن توی یک صفحه رو ترجیح میدم. ولی به هر حال کاری بود که باید میکردیم و حالا خیلی هم سخت نبود. مسئله‌ی اصلی و بسیاااار قابل اونجایی بود که فایل‌های random\_number.py و make\_array.py از هم جدا بودن. این دقیقا به این معنی بود که دوستان لطف کرده بودن راه استفاده از کد

```
random_array = np.random.randint(0, 101, user_input)
```

رو به روی ما بسته بودن!! پس باید خوب فکر میکردم که چه شکلی تابع رو مثلا ۵ بار صدا کنم درحالی که نه از این کد استفاده کنم و نه از حلقه. شروع کردم به گشتن توی سایت‌های مختلف و chatgpt چند ساعتی داشتم روی کد هایی که پیدا میکردم کار میکردم و همیشه یه جای کار یه مشکل می‌خورد.

تا اینکه بالاخره به جایی رسید که تصمیم گرفتم بصورت بازگشتی بنویسم. خب اولش که یک مشکل وجود داشت. اونم این بود که وقتی توی تابع random\_number که بصورت بازگشتی خودش رو صدا میزد، تابع make\_array رو صدا میزد و توی یک متغیر ذخیره میکردم و در آخر اون رو return میکردم، فقط یک آرایه با اولین عدد رندمی که به وجود اومده بود رو بر میگردوند. درحالی که توی تابع make\_array آرایه رو دقیقا به تعداد عنصری که کاربر خواسته بود ذخیره کرده بود. خلاصه بعد از کلی بررسی به این نتیجه رسیدم که خب اوکی اصلا return نمیکنم. بعدا که خواستم ارزش استفاده می کنم. مشکل بعدی وقتی بوجود آمد که فایل app.py رو به پروژه اضافه کردم. و به نحوه import کردن من گیر میداد.

```
from make_array import make_array
```

خیلی توی اینترنت گشتم. یکی میگفت بخاطر تفاوت نسخه آنکوندا میتونه باشه و من نسخه هارو عوض کردم و باز هم اتفاقی نیوفتاد. و هیچ وقت فکر نمیکردم به خاطر یک نقطه که باید قبل make\_array میذاشتم اینقد وقتم ازتلف بشه (اینم یکی

دیگه از دلایلی که کار کردن با چند فایل رو واسه این مینی پروژه  
ها دوست ندارم!!!)

```
from .make_array import make_array
```

خب حالا وقت اون رسیده بود که از کاربر ورودی رو دریافت  
کنیم. نکته مهم این بود که ورودی کنترل بشه! برای نمایش خطا  
تصمیم گرفتم از messagebox استفاده کنم.

- کاربر نباید ورودی رو خالی می گذاشت
- کاربر باید یک مقدار عددی رو وارد میکرد
- کاربر باید اعداد بین ۰ تا یک عدد معقولی مثل ۱۰۰۰ رو  
وارد میکرد

که به شکل زیبایی کار میکنه! بعد از چیدن آپشن های مختلف به  
یک مشکل رسیدم. اینکه دفعه اول برنامه درست کار می کرد و  
اینکه تونستم بعد از اینکه

- دکمه ثبت رو که زد، ورودی رو براش خالی کنه
- ولی دفعه دوم که میخواست وارد کنه اعداد هرچیزی که برای  
دفعه قبلی نوشته شده بود پاک نمی شد و مطالب جدید میومدن  
زیرش قرار میگرفتن. و واقعا میگم که سر این مشکل ۳ رو

درگیر بودم و خیلی اذیت شدم که بالاخره یه شب ساعت ۲ صبح چاره رو یافتم! باید عناصر بچه(children elements) رو پاک میکردم. ولی باز در چند جا مورد داشت:

- در چه زمانی و در کجای برنامه از این کد استفاده کنم
  - با این کد همه عناصر از جمله محل گرفتم ورودی و دکمه ثبت هم حذف می شدند
  - باید از حلقه for استفاده می کردم که نمیخواستم!
- اینو فهمیدم که یک COUNTER بذارم واسه ی هرباری که کاربر روی دکمه ثبت کلیک میکنه. بعد بگم اگه  $COUNTER > 1$ ، یعنی حد اقل دفعه دومه که داره ثبت میکنه و این یعنی اطلاعات قبلی روی صفحه موجود هستند و باید پاک بشن. و داخل فایل app.py ازش استفاده کردم. برای مشکل حلقه، اون رو با np.vectorize و lambda جایگزین کردم و دقیقا همون کارو رو برام میکرد. و اینکه براش شرط گذاشتم که همه بچه های window رو بجز (label , entry , button) پاک کنه که در کل بصورت زیر شد:

```
if COUNTER > 1:
    np.vectorize(lambda child: child.destroy() if(child != button and child !=
keep_label and child != entry) else None)(window.wininfo_children())
```

یک مشکلی هم که این وسط پیش اومد ارور RecursionError بود.

```
'RecursionError: maximum recursion depth exceeded while calling a Python object'
```

من حالت حداکثر رو روی ۱۰۰۰ گذاشته بودم ولی این از اعداد ۹۷۷ تا ۱۰۰۰ رو RecursionError می داد. برای همین limit رو روی ۹۷۶ تنظیم کردم.

- و در آخر هم بخش امکان ثبت با کلید <ENTER> برای راحتی کاربر

بعد رفتم قسمت اکسترا رو انجام بدم که کار سخت و مشکلی نبود | اینکه با string Doc و argparse به صورت اسمی آشنا نبودم ولی با یک سرچ ساده فهمیدم چی به چیه و اینم انجام شد. ببینیم چی میشه! خسته نباشید 🌹