

به نام خدا

---

## گزارش کار تمرین دوم

---

مدرسۀ:

فیلاگر

مؤلف:

سعید قادری زاده

زمستان ۱۴۰۲

در این تمرین، با استفاده از کتابخانه NumPy در زبان برنامه‌نویسی Python، سه تابع اصلی به منظور تولید عدد تصادفی، ایجاد آرایه با اعداد تصادفی و تعیین شکست یا پیروزی طراحی شده. در ادامه، تجربیات در طراحی و اجرای این تمرین به تفصیل بررسی می‌شود.

## ۱. استفاده از کتابخانه Numpy:

- استفاده از کتابخانه NumPy به سرعت و کارایی کد و امکان ایجاد آرایه‌های تصادفی با تعداد دلخواه را به سادگی فراهم می‌کند.

## ۲. تولید عدد تصادفی:

- تابع generate\_random\_number به سادگی یک عدد تصادفی بین ۰ تا ۱۰۰ ایجاد می‌کند.

## ۳. ایجاد آرایه با اعداد تصادفی:

- تابع generate\_random\_array با دریافت اندازه مورد نظر از کاربر، یک آرایه با اعداد تصادفی ایجاد می‌کند.

## ۴. تعیین شکست یا پیروزی:

- تابع identify\_result به سادگی شرط بررسی شکست یا پیروزی را اعمال می‌کند و پیام مناسب را باز می‌گرداند.

## چالش اول:

در تمرین اول، از ویجت place برای تعیین مکان کلیدها و برچسب‌ها استفاده شد. این باعث می‌شد که با بزرگ‌تر شدن پنجره، موقعیت کلیدها ثابت بماند و تغییر نکند. اما در این برنامه از ویجت pack استفاده شده است و چالشی که در ابتدا به وجود آمد دکمه‌های Start و Exit زیر هم قرار می‌گرفتند.

## راه حل:

با استفاده از کد نوشته شده دکمه‌ها در کنار هم قرار گرفتند و با تغییر اندازه در وسط قرار می‌گیرند  
با استفاده از کد نوشته شده، دکمه‌های Start و Exit در کنار یکدیگر قرار گرفته‌اند و با تغییر اندازه برنامه در وسط قرار می‌گیرند.  
(کد ۲۳ و ۲۶)

```
self.start_button.pack(side='left', anchor='e', expand=True)

self.exit_btn.pack(side='right', padx=5, anchor='w', expand=True)
```

## چالش دوم:

بعد از وارد کردن توسط کاربر عدد باقی میماند در پنجره ورودی

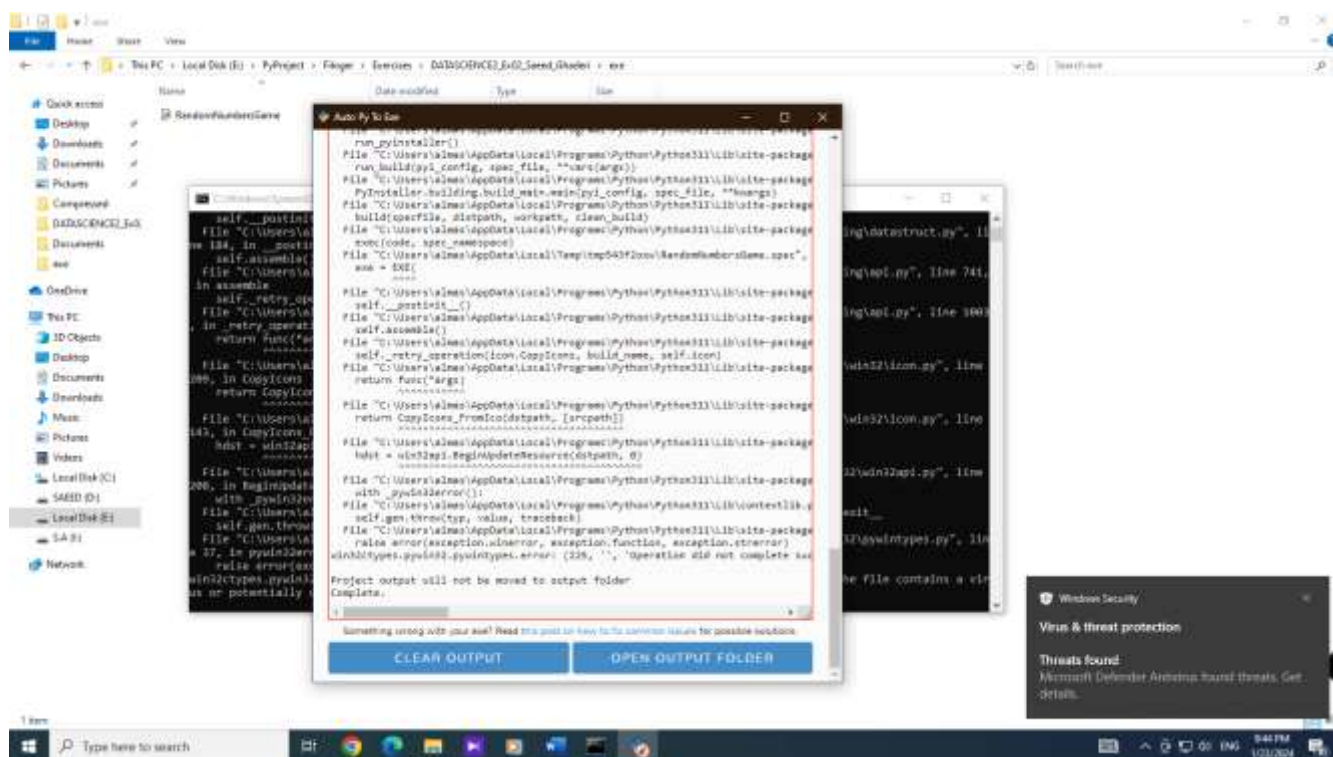
## راه حل:

برای برطرف شدن از کد زیر بعد از هر پیغام عددی که از قبل وارد شده پاک می‌شود. (کد ۳۶ و ۴۳)

```
self.entry.delete(0, tk.END)
```

## چالش سوم:

در هنگام تبدیل فایل پایتون به exe برنامه خطای زیر را نشان می‌دهد.



## راه حل:

پس از غیرفعال کردن آنتی‌ویروس و اجرای مجدد برنامه تبدیل فایل پایتون به exe، مشکل برطرف شد.

## چالش چهارم:

اجرای برنامه app در فایل extra با ورودی کاربر در ترمینال امکان‌پذیر بود و جواب مورد نظر دریافت می‌شد، اما در اجرای برنامه با ورودی کاربر به خوبی عمل نمی‌کرد.

## راه حل:

در ابتدا، اگر در ترمینال عددی وارد نمی‌شد، همچنان مقدار None در نظر گرفته شود. این امر باعث می‌شود که برنامه حتی اگر کاربر هیچ عددی وارد نکند، نیز اجرا شود. به همین دلیل، یک تابع جداگانه به نام `get_user_input()` برای ورودی گرفتن از کاربر در برنامه اضافه شد. (کد ۲۵-۲۹)

```
if args.number is not None:
    user_number = args.number

else:
    user_number = get_user_input()
```