



KAPREKAR 6174

winter 1404

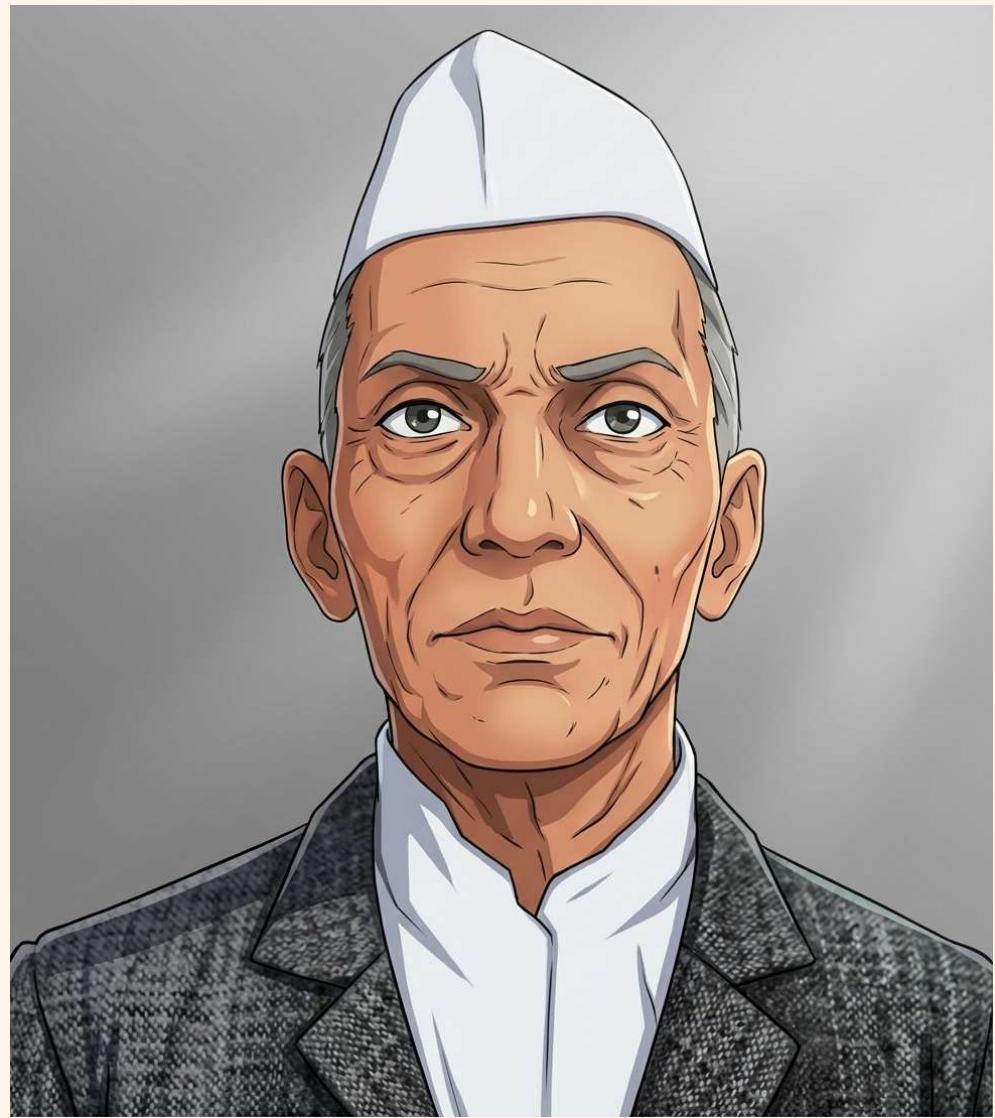
Somayeh Hosseinbeig
info@somayehhosseinbeig.com



دی. آر. کاپر کار

۱۹۰۵—۱۹۸۶

ریاضی دان هندی



عملیات کاپرکار

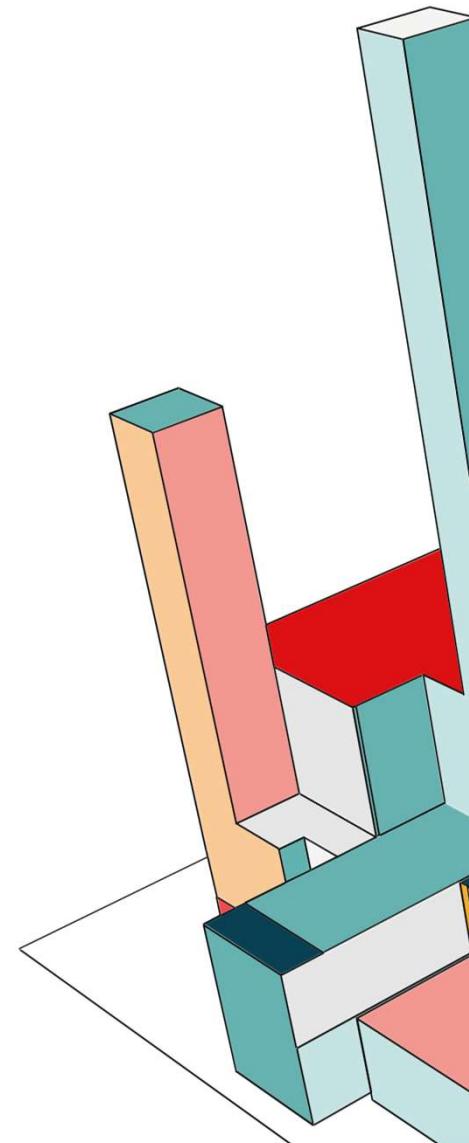
عملیات کاپرکار یک فرایند ساده و جالب در دنیای اعداد است که نشان می‌دهد چگونه می‌توان با تکرار چند مرحله‌ی منظم و مشخص، به یک الگوی پایدار رسید. در این روش، با جابه‌جا کردن رقم‌ها و انجام یک عمل تکراری، عدد به سمتی حرکت می‌کند که در نهایت به یک حالت ثابت می‌رسد. جذابیت عملیات کاپرکار در این است که بدون استفاده از فرمول‌های پیچیده، مفهوم نظم، تکرار و همگرایی را به شکلی شهودی و قابل فهم نشان می‌دهد و برای آشنایی اولیه با تفکر الگوریتمی بسیار مناسب است.



عملیات کاپرکار

ایدهی اصلی

ما یک عدد ۴ رقمی انتخاب می‌کنیم (به شرطی که همهی رقم‌هایش یکسان نباشند، مثل ۱۱۱۱). بعد یک سری کار تکراری روی آن انجام می‌دهیم تا در نهایت به عدد ۶۱۷۴ برسیم که به آن ثابت کاپرکار می‌گویند. مراحل عملیات کاپرکار در اسلاید بعدی خواهیم دید



عملیات کاپرکار

مراحل عملیات کاپرکار

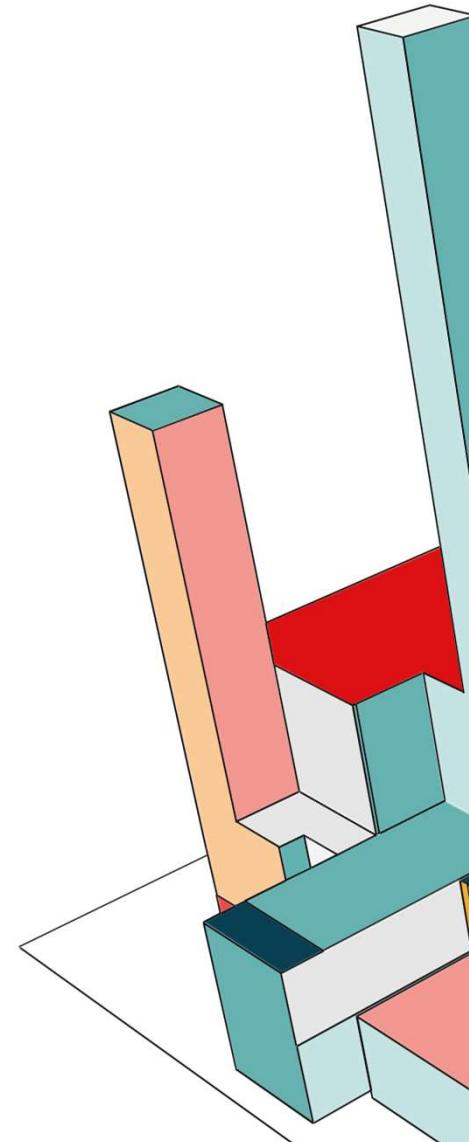
فرض کن عدد ما ۳۵۲۴ است:

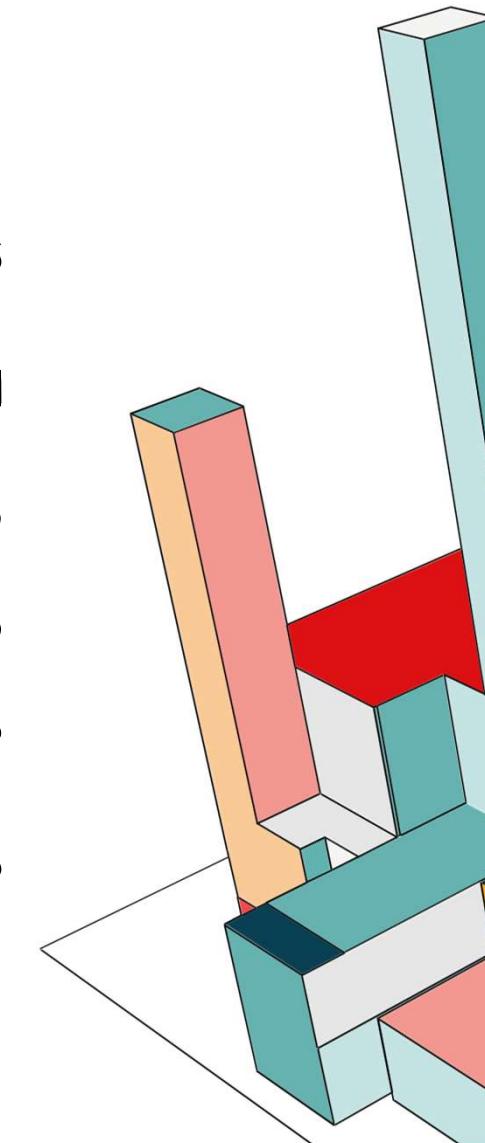
مرتبسازی رقمها به صورت نزولی (بزرگ به کوچک) ۵۴۳۲

مرتبسازی رقمها به صورت صعودی (کوچک به بزرگ) ۲۳۴۵

تفریق عدد کوچک از عدد بزرگ $5432 - 2345 = 3087$

حالا عدد جدید (۳۰۸۷) را دوباره وارد همین مراحل می‌کنیم.





عملیات کاپرکار

نکته مهم

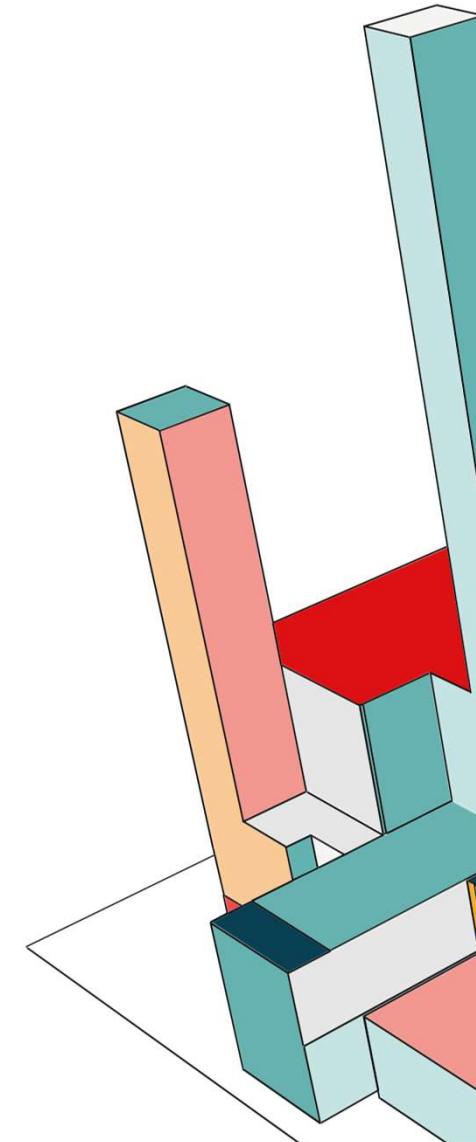
این روند تقریباً برای همهی عددهای ۴ رقمی (به جز عددهایی که همه رقم‌هایشان یکی است) در چند مرحله به ۶۱۷۴ می‌رسد. وقتی به ۶۱۷۴ رسیدیم، اگر دوباره مراحل را انجام بدھیم، همیشه دوباره ۶۱۷۴ به دست می‌آید.

به همین دلیل به ۶۱۷۴ می‌گویند ثابت کاپرکار.

عملیات کارپرکار

```
def kaprekar(num):
    largest = ''.join(sorted(num, reverse=True))
    smallest = ''.join(sorted(num))
    difference = int(largest) - int(smallest)
    print(f"Largest number: {largest}")
    print(f"Smallest number: {smallest}")
    print(f"Difference: {difference}")
    print("-----")
    return difference
```

ابتدا یک تابع تعریف می کنیم برای عملیات کارپرکار یک عدد ورودی میگیرد سپس دو متغیر بزرگتر و کوچکتر می سازد و آنها را از هم کم می کند و نمایش می دهد



عملیات کا پرکار

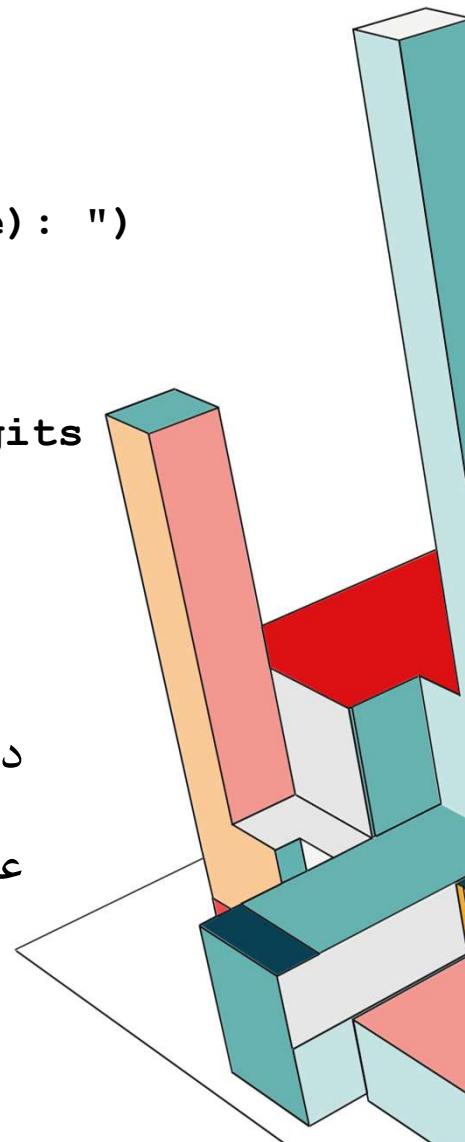
```
num = input("Enter a 4-digit number (not all digits the same) : ")

if len(num) != 4 or len(set(num)) == 1 or not num.isdigit():

    print("Invalid input. Please enter a 4-digit number (digits
only) and not all digits the same (e.g., 1111).")

    exit()
```

در این قسمت از کاربر ورودی را می گیریم و چک می کنیم که حتما عدد باشد و تمامی ارقام آن یکسان نباشد و تعداد ارقام آن ۴ رقم باشد



عملیات کاپرکار

```
step = 0

while num != "6174":

    step += 1

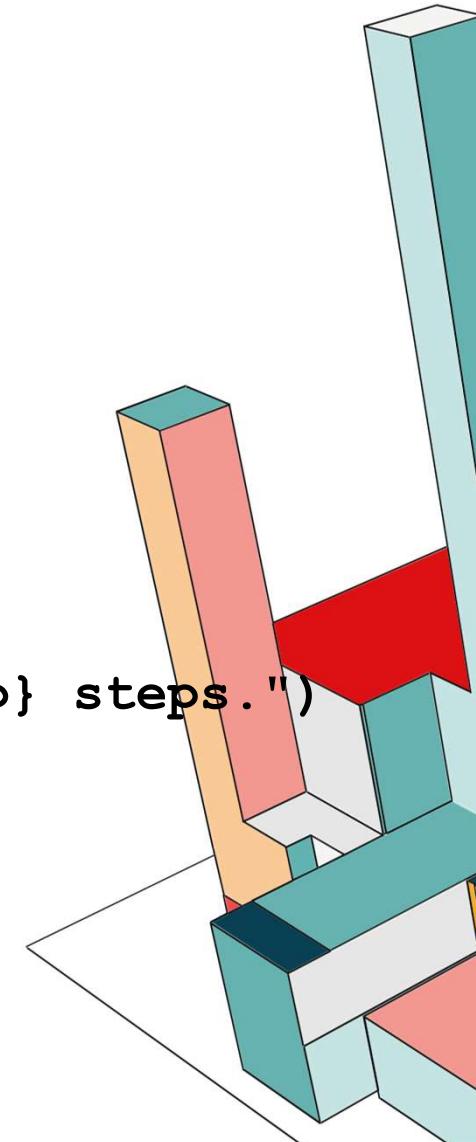
    print(f"Step {step}:")
    num = str(kaprekar(num)).zfill(4)
    print(f"Next number: {num}")

print(f'Reached Kaprekar's constant 6174! in {step} steps.')
```

در اینجا ابتدا یک متغیر برای تعداد دفعات تعریف می کنیم سپس درون

حلقه تا زمانی که به عدد ۶۱۷۴ نرسیم ادامه می دهیم اگر عدد ما تعداد

ارقامش از ۴ کمتر بود به آن صفر می افزاییم



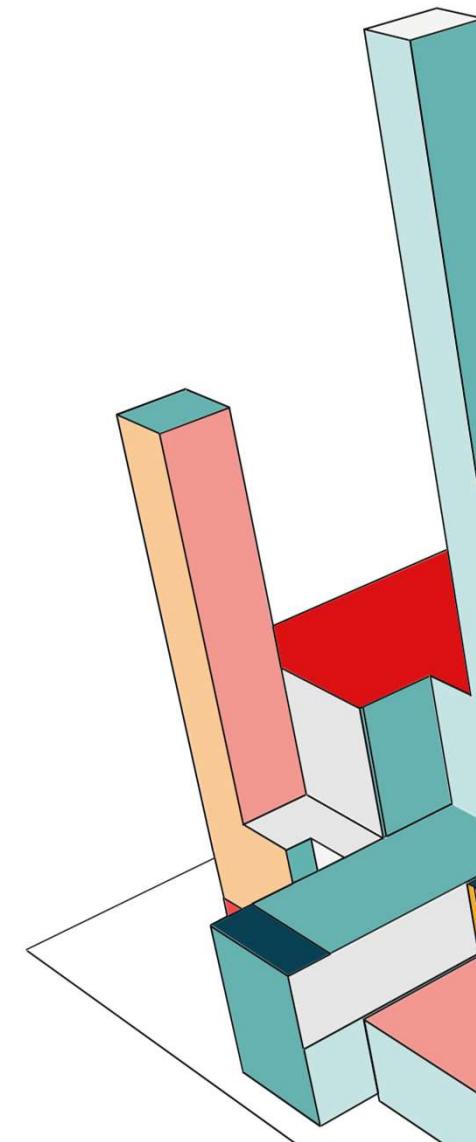
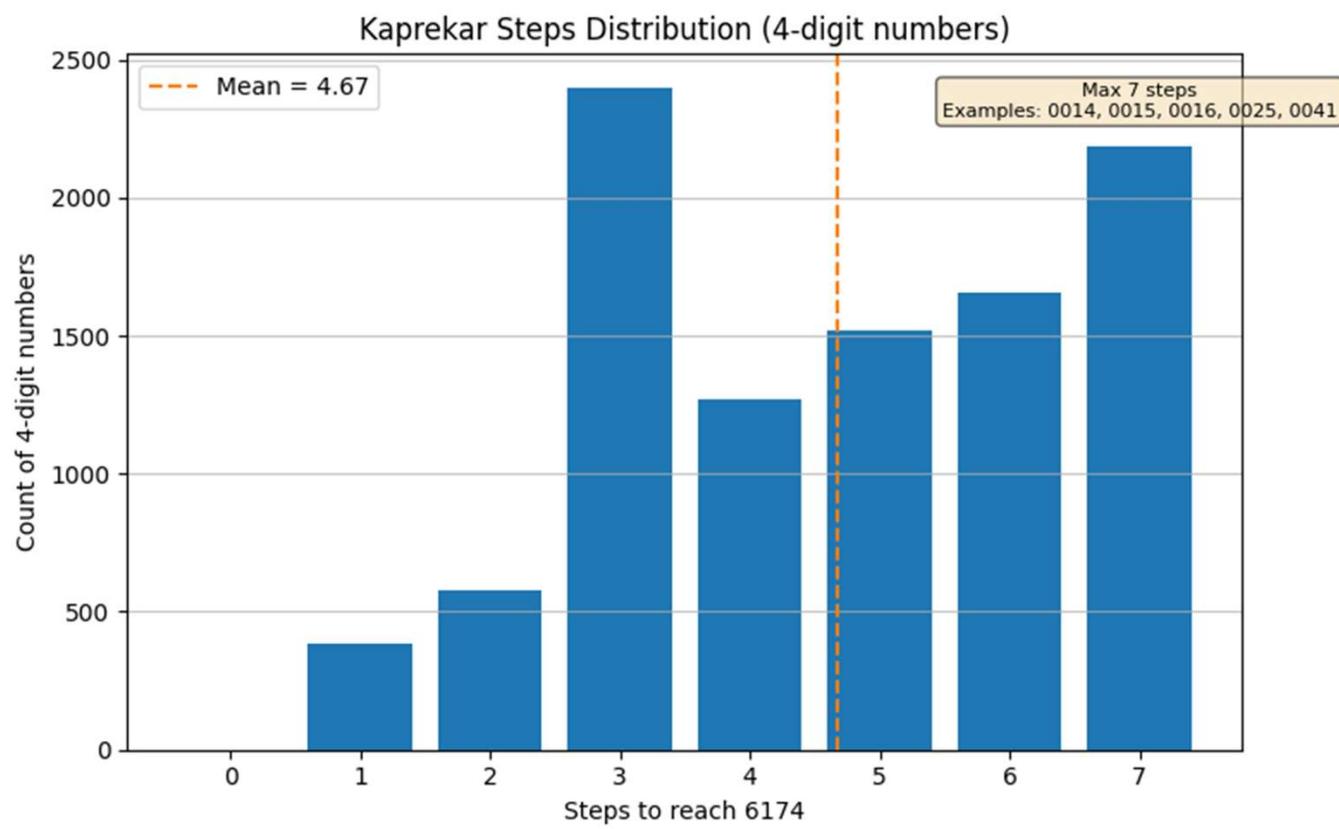
تمرین کلاسی

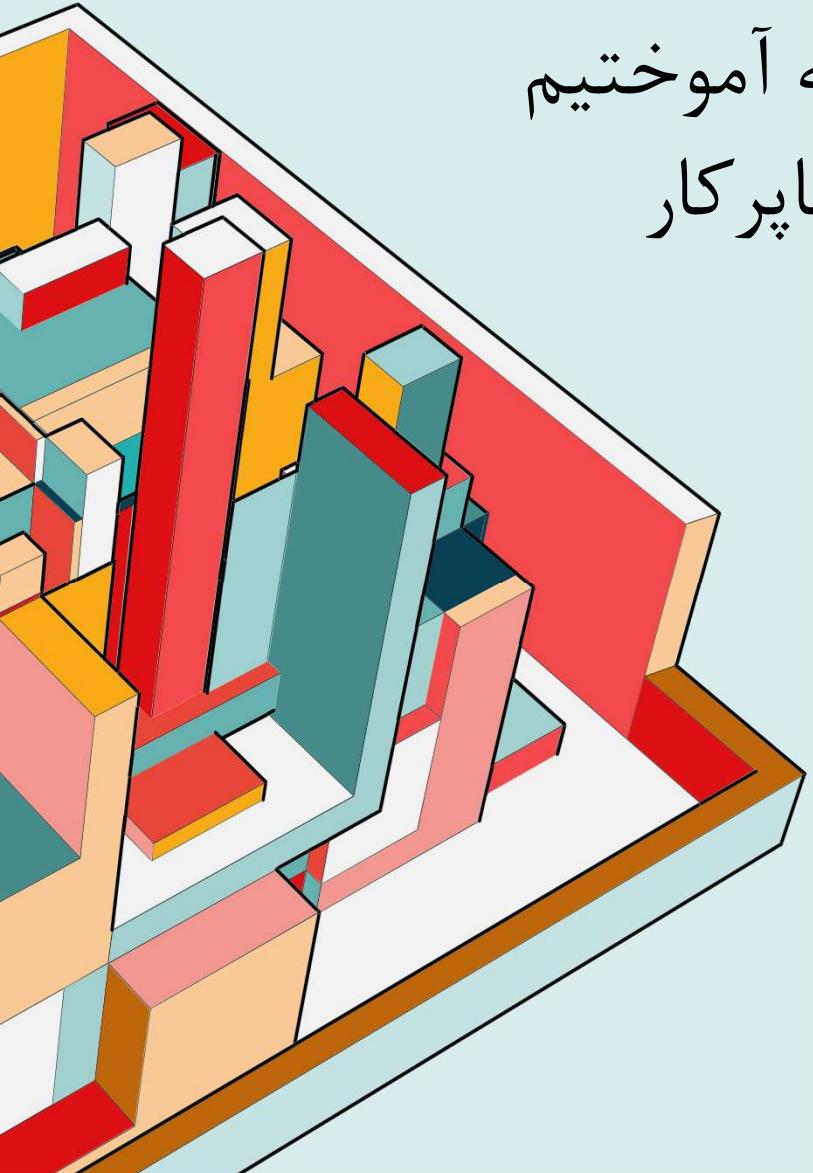
برنامه را توسعه دهید که بتواند تمامی اعداد ۴ رقمی را تست کند.

- الف) میانگین step را محاسبه نمایید
- ب) بیشینه step را محاسبه نمایید
- ج) مقادیر step را ذخیره نموده و هیستوگرام آن را رسم کنید.



هیستوگرام تعداد دفعات





در این جلسه آموختیم

۱- عملیات کاپرکار



THANK YOU

Ask questions if any!

