املاکی آقای آب‌دماغ

توجه: در این سوال مجاز به استفاده از کتابخانه پانداز (pandas) **نیستید**!

آقای آب‌دماغ که به تازگی تمایل به سرمایه‌گذاری در بازار املاک پیدا کرده تا با سود آن شرکتش را گسترش دهد، برای پردازش داده‌های ملکی از سالیوان و مایک کمک خواسته. آقای آب‌دماغ یک جدول با بیش از ۳۰۰۰ ردیف به آن‌ها داده و از آن‌ها خواسته تا یک ابزار فیلترنویسی برایش بسازند.

در این سوال یک فایل به نام `housePrice.csv` به شما داده خواهد شد و شما باید اطلاعات مربوطه را از آن استخراج نمایید.

ساختار جدول و فایل آن

چند سطر اول این جدول در ادامه آمده است:

Area	Room	Parking	Warehouse	Elevator	Address	Price
63	1	True	True	True	Shahran	1850000000

Area	Room	Parking	Warehouse	Elevator	Address	Price
60	1	True	True	True	Shahran	1850000000
79	2	True	True	True	Pardis	550000000
95	2	True	True	True	Shahrake Qods	902500000
123	2	True	True	True	Shahrake Gharb	7000000000
70	2	True	True	False	North Program Organization	2050000000
87	2	True	True	True	Pardis	600000000
59	1	True	True	True	Shahran	2150000000
54	2	True	True	False	Andisheh	493000000
71	1	True	True	True	West Ferdows Boulevard	2370000000
68	2	True	True	True	West Ferdows Boulevard	2450000000

که این جدول در فایل CSV مربوطه، متناظر با کد زیر است:

```
Area,Room,Parking,Warehouse,Elevator,Address,Price
63,1,True,True,True,Shahran,1850000000
60,1,True,True,True,Shahran,1850000000
79,2,True,True,True,Pardis,550000000
95,2,True,True,True,Shahrake Qods,902500000
123,2,True,True,True,Shahrake Gharb,7000000000
70,2,True,True,False,North Program Organization,2050000000
87,2,True,True,True,Pardis,600000000
59,1,True,True,True,Shahran,2150000000
54,2,True,True,False,Andisheh,493000000


```

```
71,1,True,True,True,West Ferdows Boulevard,2370000000
68,2,True,True,True,West Ferdows Boulevard,2450000000
```

کاری که باید انجام دهید

یک تابع به نام `filter_houses` می‌نویسید که تنها آرگومانش یک دیکشنری است، یعنی:

```python
def filter_houses(filters: dict):
    # کد شما
```

فیلتر می‌تواند کلیدهای زیر را داشته باشد:

نام کلید	نوع مقادیر مجاز برای کلید	توضیح مقدار
area	عدد صحیح (`int`)	فیلتر برای ردیف‌هایی که متراژ خانه (Area) در آن‌ها **حداقل** برابر با عدد داده شده است.
room	عدد صحیح (`int`)	فیلتر برای ردیف‌هایی که تعداد اتاق خواب‌ها (Room) در آن‌ها برابر با عدد داده شده است.
parking	مقدار منطقی (`bool`)	فیلتر برای ردیف‌هایی که پارکینگ (Parking) داشتن یا نداشتنشان (True/False) مطابق با آنچه داده شده است.
warehouse	مقدار منطقی (`bool`)	فیلتر برای ردیف‌هایی که انباری (Warehouse) داشتن یا نداشتنشان (True/False) مطابق با آنچه داده شده است.
elevator	مقدار منطقی (`bool`)	فیلتر برای ردیف‌هایی که آسانسور (Elevator) داشتن یا نداشتنشان (True/False) مطابق با آنچه داده شده است.
address	رشته (`str`)	فیلتر برای ردیف‌هایی که آدرسشان (Address) مطابق رشته‌ی داده شده است.
price	عدد صحیح (`int`)	فیلتر برای ردیف‌هایی که قیمتشان (Price) **حداکثر** برابر با مقدار داده شده است.

در نهایت تابع باید ردیف‌هایی که فیلتر شده‌اند را در یک فایل به نام `results.csv` ذخیره کند. توجه کنید که سطر اول فایل CSV شما باید به صورت زیر باشد:

```
Area,Room,Parking,Warehouse,Elevator,Address,Price
```

در سطرهای بعد نیز باید ردیف‌هایی بیایند که با فیلتر جدا کردید.

توجه: برای راحتی کار شما، تضمین می‌شود تمام کلیدهای داخل دیکشنری فیلتر از حروف کوچک لاتین تشکیل شده باشند.

ورودی و خروجی

این سوال ورودی و خروجی ندارد و داور کوئرا صرفاً تابعی که می‌نویسید را داوری می‌کند.

مثال

اگر قطعه کد زیر را در انتهای کد خود اضافه کنید:

```
filters = {
    "area": 81,
    "rooms": 2,
    "elevator": True,
    "parking": True,
    "warehouse": True,
    "district": "Shahryar",
    "price": 5800000000
}

filter_houses(filters)
```

باید محتوای فایل `result.csv` در انتهای اجرای برنامه به صورت زیر باشد:

```
Area,Room,Parking,Warehouse,Elevator,Address,Price
85,2,True,True,True,Shahryar,1419000000
```

```
102,2,True,True,True,Shahryar,1225000000
127,2,True,True,True,Shahryar,2500000000
```

که این، معادل جدول زیر است:

Area	Room	Parking	Warehouse	Elevator	Address	Price
85	2	True	True	True	Shahryar	1419000000
102	2	True	True	True	Shahryar	1225000000
127	2	True	True	True	Shahryar	2500000000

راهنمایی‌ها

▼ راهنمایی ۱

می‌توانید فیلترها را به چند تابع بشکانید و بعداً از توابع استفاده کنید. مثلاً:

```python
def filter_by_area(area: int, lines: list):
# کد شما

def filter_by_rooms(rooms: int, lines: list):
# کد شما

def filter_by_parking(parking: bool, lines: list):
# کد شما

def filter_by_warehouse(warehouse: bool, lines: list):
# کد شما

def filter_by_elevator(elevator: bool, lines: list):
# کد شما

def filter_by_district(district: str, lines: list):
# کد شما


```

```
20 def filter_by_price(price: int, lines: list):
    # کد شما
```

▼ راهنمایی ۲

می‌توانید از توابع فیلتری که نوشتید را با ساختار شرطی ترکیب کرده و از منطقی شبیه کد زیر استفاده کنید:

```
1  def filter_houses(filters: dict):
2      ...
3      if "area" in filters:
4          current_rows = filter_by_area(filters["area"], current_r
5      if "rooms" in filters:
6          current_rows = filter_by_rooms(filters["rooms"], current_
7      if "parking" in filters:
8          current_rows = filter_by_parking(filters["parking"], curr
9      if "warehouse" in filters:
10         current_rows = filter_by_warehouse(filters["warehouse"],
11     if "elevator" in filters:
12         current_rows = filter_by_elevator(filters["elevator"], cu
13     if "district" in filters:
14         current_rows = filter_by_district(filters["district"], cu
15     if "price" in filters:
16         current_rows = filter_by_price(filters["price"], current_
17     ...
```

بیشتر بدانید

▼ مطالعه‌ی آزاد: پردازش داده

پردازش داده شامل روش‌ها و تکنیک‌هایی است که برای جمع‌آوری، تمیزکاری، تجزیه‌وتحلیل و استخراج اطلاعات ارزشمند از داده‌ها استفاده می‌شود. در برنامه‌نویسی، این فرآیند معمولاً شامل خواندن داده‌ها از منابع مختلف (مثل فایل‌های CSV، پایگاه‌های داده، یا APIها)، اعمال تغییرات لازم روی داده‌ها (مانند حذف داده‌های ناقص یا اعمال فیلترها)، و در نهایت ذخیره‌سازی یا نمایش نتایج است. ابزارها و کتابخانه‌های

متنوعی مانند Pandas در Python، نقش مهمی در ساده‌تر کردن این فرآیند دارند، اما استفاده از راهکارهای دستی و بدون وابستگی به این کتابخانه‌ها نیز برای موارد خاص، ضروری است.

یکی از چالش‌های پردازش داده، تطبیق داده‌ها با فیلترها و محدودیت‌های تعیین‌شده است. این کار نیازمند درک درست از ساختار داده‌ها و نحوه اعمال عملیات منطقی بر روی آن‌هاست. با استفاده از ساختارهایی مانند لیست‌ها و دیکشنری‌ها در زبان‌های برنامه‌نویسی، می‌توان پردازش داده‌ها را به صورت دستی انجام داد و خروجی‌های دلخواه را تولید کرد. این نوع پردازش به‌خصوص زمانی که محدودیت‌هایی بر روی استفاده از کتابخانه‌های خاص وجود دارد، اهمیت بیشتری پیدا می‌کند.

پیشنهاد می‌شود اگر به این شاخه علاقه دارید، دوره‌های سایت Kaggle را بگذرانید.

▼ مطالعه‌ی آزاد: فیلتر کردن داده‌ها در دنیای واقعی

در دنیای واقعی، فیلتر کردن داده‌ها کاربردهای گسترده‌ای دارد و تقریباً در هر حوزه‌ای که داده‌ها تولید می‌شوند، مورد استفاده قرار می‌گیرد. از جمله این حوزه‌ها می‌توان به تحلیل بازار، مراقبت‌های بهداشتی، تجارت الکترونیک، حمل‌ونقل و شبکه‌های اجتماعی اشاره کرد. هدف اصلی فیلتر کردن داده‌ها، انتخاب زیرمجموعه‌ای از داده‌هاست که نیازهای خاص کاربران یا سیستم‌ها را برآورده می‌کند. برای مثال، در تجارت الکترونیک، کاربران اغلب از فیلترهایی مانند قیمت، دسته‌بندی محصول، برند، و نظرات مشتریان برای پیدا کردن محصولاتی که نیاز دارند استفاده می‌کنند. همچنین در مراقبت‌های بهداشتی، پزشکان و محققان ممکن است داده‌های بیمارانی با شرایط خاص را فیلتر کنند تا بتوانند روندهای بیماری یا نتایج درمان‌ها را بررسی کنند. در حوزه تحلیل بازار، شرکت‌ها از فیلترهایی بر اساس موقعیت جغرافیایی، سن، و رفتار خرید برای تعیین استراتژی‌های بازاریابی خود بهره می‌برند. فیلتر کردن داده‌ها همچنین در سیستم‌های توصیه‌گر (مانند نتفلیکس یا اسپاتیفای) نقش کلیدی دارد. این سیستم‌ها از الگوریتم‌های پیچیده و داده‌های فیلتر شده برای ارائه پیشنهادات شخصی‌سازی‌شده به کاربران استفاده می‌کنند. به طور کلی، فیلتر کردن داده‌ها ابزاری ضروری برای تبدیل حجم وسیع داده‌ها به اطلاعات معنادار و قابل استفاده است.

▼ مطالعه‌ی آزاد: سیستم‌های پیشنهاددهی

سیستم‌های پیشنهاد دهی ابزارهایی هستند که به کاربران پیشنهادات شخصی‌سازی‌شده می‌دهند تا تجربه کاربری بهتری ایجاد کنند. این سیستم‌ها معمولاً بر اساس داده‌های رفتار گذشته کاربر یا کاربران دیگر عمل

می‌کنند. مثلاً در سایت‌های خرید آنلاین، سیستم پیشنهاددهی ممکن است محصولاتی مشابه به خریدهای گذشته یا علایق کاربران را پیشنهاد دهد.

سه نوع اصلی سیستم پیشنهادهی وجود دارد: **مبتنی بر فیلتر محتوایی**، که پیشنهادات را بر اساس ویژگی‌های محصولات می‌دهد، **مبتنی بر فیلترسازی هم‌زمان**، که پیشنهادات را بر اساس رفتار کاربران مشابه ایجاد می‌کند، و **ترکیبی** که از ترکیب هر دو روش قبلی استفاده می‌کند تا پیشنهادات دقیق‌تری بدهد. سیستم‌های ترکیبی معمولاً دقت بیشتری دارند چون از مزایای هر دو روش بهره می‌برند.

▼ مطالعه‌ی آزاد: پیش‌بینی آینده، ممکن یا ناممکن؟

پیش‌بینی آینده در زمینه علوم داده به معنای استفاده از داده‌های موجود و الگوریتم‌های پیچیده برای پیش‌بینی اتفاقات یا روندهای آینده است. در این حوزه، با تحلیل داده‌های تاریخی و شبیه‌سازی سناریوهای مختلف، می‌توان مدل‌هایی ساخت که احتمالات مختلف را محاسبه کنند. برای مثال، در پیش‌بینی قیمت سهام، رفتار کاربران یا وضعیت آب‌وهوا، الگوریتم‌های یادگیری ماشین مانند رگرسیون، شبکه‌های عصبی و درخت تصمیم می‌توانند به‌طور نسبی پیش‌بینی‌هایی دقیق ارائه دهند. با این حال، پیش‌بینی آینده در علوم داده همیشه با چالش‌هایی همراه است. داده‌ها ممکن است ناقص، نادرست یا پراکنده باشند (مثلا نویز داشته باشند!) و همین موضوع دقت پیش‌بینی‌ها را کاهش می‌دهد. همچنین، برخی عوامل غیرقابل پیش‌بینی مانند تغییرات ناگهانی در بازار بورس یا رخدادهای طبیعی مانند زلزله می‌توانند پیش‌بینی‌ها را با خطا مواجه کنند. به همین دلیل، همیشه باید به پیش‌بینی‌ها با دقت و در نظر گرفتن احتمالی بودن نگاه کرد. می‌توان با استفاده از روش‌ها و ابزارهای موجود، تصمیمات هوشمندانه‌تری گرفت و احتمالات مختلف را بهتر درک کرد. اما این لزوماً به معنی پیش‌بینی آینده نخواهد بود!