


## فاکتوریسم

در این سوال شما باید یک کلاس مدیریت فاکتورهای فروش برای شرکت سجادینه طراحی کنید. در شرکت سجادینه افراد زیادی می‌توانند فاکتورهای مالی را ثبت کنند و هر کدام از این افراد تاریخ فاکتور را به یکی از ۶ فرمتی که در جدول پایین آمده ثبت می‌کنند.

برای مثال یک فاکتور که روز دوم ماه ششم سال ۲۰۲۰ ثبت شود، می‌تواند طبق یکی از فرمت‌های زیر ثبت شود:

| فرمت زمان  | زمانی که در سیستم ثبت می‌شود |
|------------|------------------------------|
| dd/mm/yyyy | 02/06/2020                   |
| dd/yyyy/mm | 02/2020/06                   |
| yyyy/mm/dd | 2020/06/02                   |
| yyyy/dd/mm | 2020/02/06                   |
| mm/yyyy/dd | 06/2020/02                   |
| mm/dd/yyyy | 06/02/2020                   |

شما باید کلاس FactorHandler را در این فایل کامل کنید به طوری که سه تابع زیر عملکردهای خواسته شده را داشته باشند (عملکرد تابع \_\_init\_\_ اهمیت ندارد).

 solution.py

```
class FactorHandler:
    def __init__(self):
        pass

    def add_factor(self, time_format, time, value):
        pass

    def remove_all_factors(self, time_format, time):
        pass
```

```

10 |
11 |     def get_sum(self, time_format, start_time, finish_time):
12 |         pass

```

#### ▼ تابع add\_factor

این تابع به ترتیب سه ورودی `time_format` و `time` و `value` را ورودی می‌گیرد. مقدار `time_format` یکی از ۶ رشته جدول بالا است و `time` تاریخ ثبت فاکتور بر حسب فرمت داده شده است. مقدار `value` هم، هزینه فروش فاکتور داده شده برای آن تاریخ را مشخص می‌کند.

#### ▼ تابع remove\_all\_factors

این تابع به ترتیب سه ورودی `time_format` و `time` را ورودی می‌گیرد که هر دویشان مانند تابع `add_factor` هستند؛ این تابع تمام فاکتورهایی که در تاریخ داده شده ثبت شده‌اند را از لیست فاکتورها حذف می‌کند.

#### ▼ تابع get\_sum

این تابع به ترتیب سه ورودی `time_format` و `start_time` و `finish_time` را ورودی می‌گیرد و جمع هزینه فاکتورهایی که از تاریخ `start_time` تا `finish_time` ثبت شده‌اند (شامل خود این دو تاریخ) را برمی‌گرداند.

در همه توابع تضمین می‌شود که `time` حتما بر حسب فرمت `time_format` است. توجه کنید که هیچ یک از توابع بالا را نباید استاتیک تعریف کنید.

## نمونه

### terminal

```

>>> fh = FactorHandler()
>>> fh.add_factor("dd/mm/yyyy", "02/10/2019", 10)
>>> fh.add_factor("dd/mm/yyyy", "03/10/2019", 20)
>>> fh.add_factor("dd/mm/yyyy", "03/10/2019", 30)
>>> fh.add_factor("dd/mm/yyyy", "05/10/2019", 5)
>>> fh.get_sum("yyyy/dd/mm", "2019/02/10", "2019/03/10")
60

```

```
9 | >>> fh.remove_all_factors("mm/dd/yyyy", "10/03/2019")
10 | >>> fh.get_sum("yyyy/dd/mm", "2019/02/10", "2019/05/10")
    15
```

## نکات

- می‌توانید فایل اولیه این سوال را از این [لینک](#) دریافت کنید.
- تابع `get_sum` نباید مقداری را چاپ کند، بلکه باید مقادیر مورد نظر را بازگرداند.

## نحوه ارسال

شما باید کلاس `FactorHandler` موجود در فایل `solution.py` را تکمیل و سپس این فایل را ارسال کنید.