



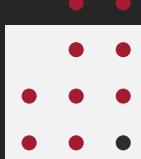
Grocery List

Problem set



فرض کنید شما عادت دارید لیستی از کالاهای فروشگاه مواد غذایی که به آن نیاز دارید، تهیه کنید.

در فایلی به نام `grocery.py`، برنامه‌ای بنویسید که از کاربر آیتم‌های موردنیاز را خط به خط تا جایی که کاربر دکمه `Ctrl + d` را فشار دهد، دریافت کند. سپس پس از فشردن `Ctrl + d` لیست آیتم‌های مواد غذایی را به صورتی در خروجی نشان دهد که همه حروف آن بزرگ باشد، به ترتیب حروف الفبا مرتب شده باشد و قبل از اسم هر آیتم، تعداد دفعاتی که کاربر آن آیتم را وارد کرده، نشان دهد. برای اسم آیتم‌ها نیازی نیست از نشانگرهای جمع استفاده کنید. بدون حساسیت به بزرگ و کوچک بودن حروف ورودی کاربر، با آن برخورد کنید.



نکته ▼

- دقت داشته باشید شما با زدن کلید ترکیبی `Ctrl + d` با خطای `EOFError` مواجه می‌شوید. با نوشتن کدی مانند زیر می‌توانید این خطا را مدیریت کنید:

```
try:  
    item = input("Item: ")  
except EOFError:  
    ...
```

- ممکن است بخواهید یک خط جدید چاپ کنید تا نشانگر کاربر (و پیام بعدی) روی همان خطی که درخواست برنامه شماست باقی نماند. کد خود را به صورت زیر تغییر دهید تا بعد از پایان برنامه، نشانگر به خط بعد منتقل شود:

```
try:  
    item = input("Item: ")  
except EOFError:  
    exit("")
```



نکته ▼

- به یاد داشته باشید که dictionary توابع مختلفی دارد. توضیحات بیشتر را در اینجا می‌توانید ببینید:

docs.python.org/3/library/stdtypes.html#mapping-types-dict

- یکی از آن توابع get است که شما با نوشتن `d[key]` می‌توانید به آیتم‌های درون یک دیکشنری دسترسی داشته باشید با فرض این که `d` یک دیکشنری و `key` کلید آن آیتم مدنظر است. برای مثال به کد زیر توجه کنید:

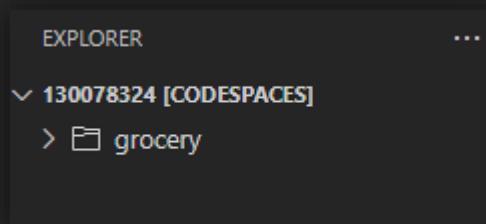
```
if key in d:  
    item = d[key]
```

وارد code.cs50.io شوید، سپس بر روی پنجره Terminal کلیک کنید. توجه داشته باشید که دستور پنجره Terminal شما باید به صورت زیر باشد:

\$

سپس کد زیر را اجرا کنید تا یک پوشه به نام grocery در codespace ایجاد شود:

\$ mkdir grocery

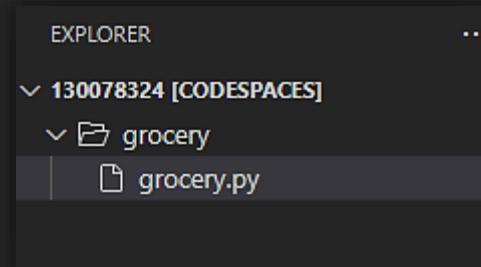


سپس `grocery` را اجرا کنید تا به پوشه‌ی `grocery` منتقل شوید. اکنون شما باید آدرس `$/grocery` را در Terminal مشاهده کنید:

```
$ cd grocery  
grocery/ $
```

حالا می‌توانید دستور `code grocery.py` را اجرا کنید تا فایلی با نام `grocery.py` ایجاد شود:

```
grocery/ $ code grocery.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

- دستور `python grocery.py` را اجرا کنید. `mango` را تایپ کنید و کلید `Enter` را فشار دهید. سپس دوباره کلمه‌ی `strawberry` را تایپ کنید و کلید `Enter` را فشار دهید و در ادامه `Ctrl + d` را بزنید. خروجی شما برابر می‌شود با:

1 MANGO

1 STRAWBERRY

- دستور `python grocery.py` را اجرا کنید. `milk` را تایپ کنید و کلید `Enter` را فشار دهید. سپس دوباره کلمه‌ی `k` را تایپ کنید و کلید `Enter` را فشار دهید و در ادامه `Ctrl + d` را بزنید. خروجی شما برابر می‌شود با:

2 MILK

- دستور `python grocery.py` را اجرا کنید. `tortilla` را تایپ کنید و کلید `Enter` را فشار دهید. سپس دوباره کلمه‌ی `sweet potato` را تایپ کنید و کلید `Enter` را فشار دهید و در ادامه `Ctrl + d` را بزنید. خروجی شما برابر می‌شود با:

1 SWEET POTATO

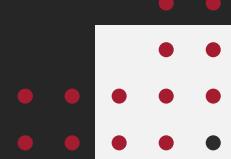
1 TORTILLA



- ❖ شما می‌توانید از آدرس زیر برای بررسی کردن کد خود استفاده کنید. CS50 از این برنامه برای آزمایش کد شما استفاده می‌کند. از این دستور استفاده کنید تا کدهایتان را امتحان کنید.

```
$ check50 cs50/problems/2022/python/grocery
```

لبخند های سبز به این معنی هستند که برنامه‌ی شما در تست قبول شده و اخم های قرمز یعنی برنامه‌ی شما دارای ایراد هست. با مراجعه به URL check50 می‌توانید خروجی مورد انتظار و خروجی برنامه‌ی خود را بررسی کنید.



با اجرا کردن صورت زیر در Terminal پاسخ خود را ارسال کنید.

```
$ submit50 cs50/problems/2022/python/grocery
```

CS50x Iran

Harvard's Computer Science 50x Iran

