

## Steve: Take the Fruit

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۵۱۲ مگابایت

والدین استیو امسال تصمیم گرفته‌اند به مناسب سال نو یک مهمانی برگزار کنند. در روز موعود، استیو که مشغول انجام پروژه‌ی برنامه‌نویسی عیدش است با آمدن میهمان‌ها فراموش می‌کند لپ‌تاپش را قفل کند. از قضا، آرتور پسرخاله‌ی چهار ساله‌ی استیو نیز میان میهمان‌هاست و زمانی که استیو مشغول پذیراییست، لپ‌تاپ او را می‌بیند. آرتور شیطان و کنجکاو، کار با لپ‌تاپ را بلد نیست و فقط شروع به بازی با موس می‌کند. خوشبختانه استیو که فردی محتاط است از قبل امکان حذف فایل را در لپ‌تاپش غیر فعال کرده بوده است. همچنین به علت وسواسش نام هر فایل را جدا از مکان قرارگیریش یکتا تعیین کرده و برای به هم نریختنشان تغییر نام فایل‌ها را نیز قفل کرده است. بنابراین آرتور با این طرف آن طرف کشیدن و کلیک بر فایل‌ها، تنها می‌تواند آن‌ها را کپی و جابه‌جا کند.

زمانی که استیو پس از میهمانی کار آرتور را می‌بیند، مجبور می‌شود پروژه‌اش را کنار گذاشته و به کمک پایتون که به تازگی شروع به یادگیری آن کرده است، مشکلش را حل کند. برای او، جابه‌جایی و مسیر فایل‌ها اهمیتی ندارد زیرا نام فایل‌ها برایش گویا و یکتا هستند. تنها دغدغه‌ی او حذف فایل‌های تکراری و آزادسازی حافظه است. در این راستا، او ابتدا به کمک نرم‌افزاری مسیر تمام فایل‌ها را به دست آورده است تا به کمک برنامه‌ای کوچک، تعداد و نام فایل‌های تکراری نیاز به حذف را پیدا کند. همچنین او که به تازگی کار با ساختار داده‌ی دیکشنری را یاد گرفته است، می‌خواهد از آن در برنامه‌اش استفاده کند. به علت مبتدی بودن و نیاز به چک کردن صحت برنامه‌اش، استیو از شما خواسته است شما نیز این برنامه‌ی کوچک را بنویسید و نسخه‌تان را برایش ارسال کنید.

بنابراین با استفاده از **دیکشنری** برنامه‌ای بنویسید که با دریافت تعداد فایل‌های سیستم و سپس مسیر فایل‌ها، فایل‌های با نام تکراری را پیدا کرده و در نهایت تعداد و نام هر فایل اضافه را به ترتیب الفبای نام، چاپ کند. در سیستم استیو، هر فایل با رشته‌ای به شکل زیر مشخص می‌شود که در آن پوشه‌ها و فایل‌ها با / از یکدیگر جدا شده‌اند و نام فایل‌ها شامل هر کاراکتری به جز / است.

/path/to/file/dir1/file1

**نکته:** ممکن است نرم‌افزاری که استیو برای لیست کردن مسیر فایل‌ها استفاده کرده است، فایل تکراری (مسیر دقیقاً یکسان) در ورودی به برنامه‌ی شما بدهد. در این صورت آن را یک فایل در نظر بگیرید و در محاسبات برای حذف فایل قرار ندهید. برای درک بهتر به مثال ۲ مراجعه کنید.

## بخش‌های برنامه

### ورودی

برنامه بایستی ابتدا عدد صحیح  $n$  را که بیانگر تعداد مسیر فایل‌های ورودی است دریافت کند. سپس در خطوط جداگانه به تعداد مشخص شده، رشته‌ی مسیر فایل دریافت کند. اگر طول آدرس فایل را  $l$  در نظر بگیریم، برای تعداد فایل‌ها و طول آدرس شروط زیر برقرارند.

$$0 \leq n \leq 1000$$

$$1 \leq l \leq 1000$$

### خروجی

برنامه باید تعداد فایل‌های اضافه برای هر نام فایل یکتا و همچنین نام فایل مربوطه را با یک فاصله میانشان در خروجی چاپ کند و در صورت نبود هیچ فایل تکراری، عدد صفر چاپ کند.

## مثال‌ها

### ورودی نمونه ۱

5

```
/home/steve/university/bp-project1.py
/home/steve/university/basic-programming/bp-project1.py
/home/steve/university/physics/physics-homework2.pdf
/home/steve/university/physics-homework2.pdf
/home/steve/university/math1/bp-project1.py
```

## خروجی نمونه ۱

```
2 bp-project1.py
1 physics-homework2.pdf
```

## ورودی نمونه ۲

```
4
/dir1/subdir4/videos/the-great-dictator.mp4
/dir2/docs/ai-links.txt
/dir1/subdir4/videos/the-great-dictator.mp4
/downloads/fdm/site.sql
```

## خروجی نمونه ۲

```
0
```

## ورودی نمونه ۳

```
5
/downloads/videos/to-delete-asap.mp4
/downloads/documents/math1-exercise1.pdf
/downloads/to-delete-asap.mp4
/downloads/fdm/to-delete-asap.mp4
/downloads/programs/math1-exercise1.pdf
```

## خروجی نمونه ۳

```
1 math1-exercise1.pdf
2 to-delete-asap.mp4
```

## Steve: General SSR

- محدودیت زمان: ۳ ثانیه
- محدودیت حافظه: ۵۱۲ مگابایت

استاد درس برنامه‌نویسی استیو که خود دانشمند بزرگی است و عادت به چالش کشیدن دانشجویانش را دارد، برای آن‌ها که تازگی شروع به یادگیری پایتون کرده‌اند پروژه‌ی سختی در تعطیلات عید قرار داده است. در این پروژه، دانشجویان باید برنامه‌ی بزرگ و سریعی برای کمک به حسابداری یک شرکت فروش سرور اینترنتی بنویسند. همچنین از آنجا که بر اساس دانش حال حاضر دانشجویان تنها از لیست‌ها و عملگرهای پایه، حل این مسئله سخت است، اعلام شده است که هر کس زودتر به پاسخی برسد که از تمامی دیگر دانشجویان بهینه‌تر باشد، لقب «ژنرال برنامه‌نویس» را از آن خود می‌کند.

استیو که عاشق برنامه‌نویسیست، مصمم است که لقب را به دست آورد. اما پس از فاجعه‌ی عید و به هم ریختن فایل‌هایش، استیو از انجام پروژه‌ی برنامه‌نویسی‌اش عقب افتاده است و نگران است فرصت نکند کدش را زودتر از هر کس دیگر آپلود کند. بنابراین او از شما خواسته است بخشی از برنامه را برای او بنویسید. در این بخش، چندین مجموعه از اعداد صحیح یا اعشاری که می‌توانند با هم اشتراکاتی داشته باشند در برنامه تولید شده‌اند و نیاز است استیو آن‌ها را ترکیب کرده و مجموعه‌ی بزرگ حاصل از اجتماع این مجموعه‌های ورودی (یعنی مجموعه‌ی اعضای تمامی مجموعه‌ها به جز عناصر تکراری) را به صورتی که اعضای مجموعه‌ی حاصل به ترتیب نزولی لیست شده باشند، بیابد. پیش از اینکار، به علت خطای محاسبات اعشار این اعداد و از آنجا که تنها تا یک رقم اعشار اعداد برای استیو مهم است، او میبایست ابتدا اعداد ورودی که اعشاری هستند را تا یک رقم اعشار گرد کند و اگر این اعشار صفر شد، آن را به عدد صحیح تبدیل کند، سپس به ساخت اجتماع بپردازد.

استیو که جلوتر از دوستانش به مطالعه‌ی درس پرداخته است، می‌داند راه‌حل سریعی برای سوال با استفاده از ساختار داده‌ی ست موجود است و چون استاد تنها زمان آپلود، زمان اجرا و صحت حاصل را مشاهده می‌کند، متوجه کلک استیو و استفاده از ست که خوانده‌اند، نخواهد شد. او از شما می‌خواهد در نوشتن سریع این بخش از تکلیف به او کمک کنید تا وقت از دست رفته به خاطر پسرخاله‌اش را جبران کند.

بنابراین با استفاده از *set* برنامه‌ای بنویسید که در  $n$  سطر،  $n$  مجموعه از اعداد صحیح یا اعشاری را دریافت کند. سپس مجموعه‌ی حاصل از اجتماع این  $n$  مجموعه را به گونه‌ای بیابد که:

۱. اعضا عناصر مجموعه‌های قبل باشند که تا یک رقم اعشار گرد شده‌اند.
۲. اگر اعشار گرد شده‌ی عددی صفر است آن عدد به عدد صحیح تبدیل شده باشد.
۳. مجموعه‌ی حاصل به ترتیب نزولی اعضایش را مرتب نشان دهد.

## بخش‌های برنامه

### ورودی

برنامه بایستی ابتدا عدد صحیح  $n$  را که بیانگر تعداد مجموعه‌هایی است که باید اجتماع گرفته شوند، در ورودی دریافت کند. سپس در  $n$  سطر جدا، هر مجموعه از اعداد صحیح یا اعشاری را در دریافت کند، که اعداد هر مجموعه با فاصله از هم رد یک سطر جدا شده‌اند. اگر تعداد اعضاء هر مجموعه را  $i$  در نظر بگیریم، برای  $n$  و  $l$  شروط زیر برقرارند.

$$1 \leq n \leq 1000$$

$$1 \leq l \leq 1000$$

### خروجی

برنامه باید در یک خط، مجموعه‌ی حاصل از اجتماع مجموعه‌های داده شده را به صورت نزولی چاپ کند. اعضای این مجموعه‌ی حاصل بایستی تا یک رقم اعشار گرد شده باشند و در صورت صفر بودن اعشار، به صورت عدد صحیح نمایش داده شوند.

## مثال‌ها

### ورودی نمونه ۱

```
4
1 2 4 5 6 8
5419.9413 3814.2121 64115.06
9 8 7 6 5 5 780 874440 41
99.1189 1.3393 96.9365 57558.8882 1.3393 96.9365 57558.8882
```

## خروجی نمونه ۱

874440 64115.1 57558.9 5419.9 3814.2 780 99.1 96.9 41 9 8 7 6 5 4 2 1

## ورودی نمونه ۲

2  
74.8842 455.9228 0.8607 995.3781  
36971.58 877.5361 80396

## خروجی نمونه ۲

80396 36971.6 995.4 877.5 455.9 74.9 0.9