

تقلب ممنوع!

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

شرکت *Snapp* جهت گسترش خدماتش به تازگی اولین نمایندگی خود را در شکرستان افتتاح کرده.

از آنجایی که آشنا شدن مردم شکرستان با *Snapp* جزو اولویتهای شرکت محسوب می‌شود، شرکت تصمیم می‌گیرد که قرعه‌کشی عظیمی میان تمامی مردم شکرستان برگزار کند.

به دنبال آن از تمامی مردم شکرستان دعوت می‌شود تا در قرعه‌کشی ثبت‌نام کنند. تنها لازمی شرکت در قرعه‌کشی این است که شرکت‌کننده‌ها کلمه شانس خود را روی یک کاغذ بنویسند و در جعبه بیاندازند. تا از میان آنها یک کاغذ به قید قرعه بیرون کشیده شود و به فردی که این کلمه را نوشته جایزه تعلق بگیرد.

اما ممکن است یک شرکت‌کننده به جای یک عدد کاغذ (یک کلمه شانس) تعدادی کاغذ (چندین کلمه‌ی شانس) داخل جعبه بیاندازد و بخواهد تقلب کند. نگران نباشید کارشناسان *Snapp* فرمول پیچیده‌ای برای حذف کردن کاغذهای اضافی دارند، از نظر کارشناسان *Snapp* هر دو کلمه شانس که پیشوندی برابر به طول حداقل p و پسوندی برابر به طول حداقل q داشته باشند، توسط یک فرد به داخل جعبه انداخته شده‌اند و از بین این کلمه‌ها (کلمه‌هایی که توسط یک نفر نوشته شده‌اند) تنها یک کلمه در جعبه می‌ماند و باقی کلمه‌ها حذف می‌شوند.

حال ما به شما تمامی کلمه‌های اولیه داخل جعبه را می‌دهیم و از شما تعداد نهایی کلمه‌های شانس، پس از اعمال فرمول فوق را می‌خواهیم.

تضمین می‌شود که طول هر کلمه شانس از p , q کمتر نیست.

ورودی

در اولین خط ورودی به ترتیب n و p و q به شما داده می‌شود (n برابر تعداد اولیه کلمه‌های شانس داخل جعبه است). در n خط بعدی در هر خط یک کلمه شانس (متشکل از حروف کوچک انگلیسی) به طول حداکثر ۶۰ آمده است.

$$1 \leq n \leq 20\,000$$

$$1 \leq p, q \leq 60$$

خروجی

در تنها خط خروجی تعداد کلمه‌های نهایی داخل جعبه (پس از اعمال فرمول کارشناسان) را چاپ کنید.

مثال

ورودی نمونه ۱

```
3 1 1
armin
akbar
baran
```

خروجی نمونه ۱

```
3
```

ورودی نمونه ۲

```
6 2 2
khosi
parsa
matin
ali
alli
parisa
```

خروجی نمونه ۲

```
4
```

توضیح نمونه ۲

کلمه‌های شانس parsa و parisa توسط یک نفر و کلمه‌های شانس ali و alli هم توسط یک نفر نوشته شده اند در نتیجه بعد از اعمال فرمول ۴ کلمه شانس داریم!!

جشن هدیه‌ها

- محدودیت زمان: ۳ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

زانا سالی یک بار یک جشن خاص برگزار می‌کند و تعدادی از دوستانش را به این جشن دعوت می‌کند. اسم این جشن «جشن هدیه‌ها» است! هر فردی که در این جشن شرکت می‌کند مقداری پول به همراه خود دارد و به تعدادی از دوستانش هدیه می‌دهد. روش هدیه دادن در این جشن کمی عجیب است! هر کدام از افراد یک لیست هدیه دارد که در آن لیست، نام تعدادی از دوستانش که در جشن شرکت کرده‌اند نوشته شده است و تمام پولی که همراه دارد را بین افراد این لیست به طور مساوی تقسیم می‌کند و این پول را به آنها هدیه می‌دهد! چون پول اعشاری (کوچکتر از یک) نداریم، این تقسیم‌ها تقسیم صحیح هستند و اگر تقسیم پول بین اعضای لیست باقیمانده‌ای داشته باشد، فرد هدیه دهنده این باقیمانده را برای خود نگه می‌دارد. به طور مثال اگر سائنا ۱۱ واحد پول داشته باشد و در لیست او فقط سه نفر باشند، به هر کدام از آنها ۳ واحد پول می‌دهد و ۲ واحد از پول خود را برای خود نگه می‌دارد.

حال شما برنامه‌ای بنویسید که پس از گرفتن اسامی شرکت کنندگان، مقدار پول اولیه‌ی هر کدام و لیست هدیه هر کس، مشخص کند که هرکسی چقدر سود یا زیان کرده است!

ورودی

- خط 1: عدد n که برابر است با تعداد شرکت کنندگان در جشن.
- خط 2 تا $n+1$: در هر خط اسم یکی از شرکت کنندگان.
- خط $n+1$ الی آخر: از این خط به بعد ورودی به n دسته تقسیم می‌شود که هرکدام مطابق زیر است: خط اول نام فردی که قرار است هدیه بدهد. در خط دوم دو عدد می‌آید: عدد اول مقدار پول آن فرد، عدد دوم (k) تعداد افراد موجود در لیست هدیه‌ی آن فرد در k خط بعدی در هر خط نام یکی از افراد موجود در لیست هدیه‌ی آن فرد.

می‌توانید فرض کنید نام هر دو نفر از افراد شرکت‌کننده در جشن متمایز است و

$$2 \leq n \leq 10$$

خروجی

در خروجی باید n خط چاپ کنید که در هر ابتدای هر خط نام هر شخص و بعد از آن مقدار سود او آورده شود. (اگر آن شخص ضرر کرده است، باید منفی مقدار ضرر چاپ شود.) ترتیب نام‌ها در خروجی باید مانند ترتیب نام‌ها در خطوط 2 تا $n+1$ ورودی باشد.

مثال

ورودی نمونه

```
5
dave
laura
owen
vick
amr
dave
200 3
laura
owen
vick
owen
500 1
dave
amr
150 2
vick
owen
laura
0 2
amr
vick
vick
0 0
```

خروجی نمونه

dave 302
laura 66
owen -359
vick 141
amr -150

رنگ کاری

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

محسن می‌خواهد دیوار خانه‌اش را رنگ کند، دیوار او به صورت یک جدول $n \times m$ است. برای این کار در هر مرحله سطل رنگی برمی‌دارد و یک زیر جدول مربعی از دیوار را با آن رنگ رنگ‌آمیزی می‌کند. (ممکن است خانه‌ای چندین بار رنگ‌آمیزی شود) حال او کنج‌کاو شده که تعداد رنگ‌های مختلف روی دیوار را بیابد به او در یافتن این تعداد کمک کنید!

رنگ دو خانه‌ی جدول متفاوت است اگر مجموعه رنگ‌هایی که روی آن زده شده با هم متمایز باشد همچنین توجه کنید که رنگ هر سطل با رنگ باقی سطل‌ها متفاوت است. برای درک بهتر به توضیحات مثال توجه کنید.

ورودی

در خط اول ورودی به ترتیب n, m و k آمده که نشان دهنده‌ی تعداد ردیف‌های جدول، تعداد ستون‌های جدول و تعداد سطل‌های رنگی است که محسن استفاده می‌کند.

$$1 \leq n, m, k \leq 50$$

در خط i ام از k خط بعدی به ترتیب r_i, c_i و l_i آمده که نشان دهنده‌ی شماره‌ی سطر و ستون خانه‌ی بالا چپ مربع و طول ضلع آن است.

$$1 \leq r_i \leq n$$

$$1 \leq c_i \leq m$$

$$1 \leq r_i + l_i - 1 \leq n$$

$$1 \leq c_i + l_i - 1 \leq m$$

$1 \leq l_i \leq \min(n, m)$

زیرمسئله‌ها

نمره	محدودیت‌ها
30	$1 \leq k \leq 20$
70	بدون محدودیت

خروجی

در تنها خط خروجی تعداد رنگ‌های مختلف روی دیوار را خروجی دهید.

مثال

ورودی نمونه ۱

5 5 3
1 1 3
2 2 4
1 3 3

خروجی نمونه ۱

7

شکل دیوار به صورت زیر است:

1	1	1,3	3	3
1	1,2	1,2,3	2,3	2,3
1	1,2	1,2,3	2,3	2,3
	2	2	2	2
	2	2	2	2

در هر خانه شماره سطلهایی که آن خانه توسطشان رنگ شده نوشته شده است. حال به ازای هر خانه رنگ آن را در نظر می‌گیریم (در واقع رنگ یک خانه مجموعه سطلهایی است که با آن رنگ شده) و مجموعه‌های مختلف ایجاد شده را می‌شماریم.

$\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$

ورودی نمونه ۲

3 4 3
1 1 1
2 2 2
1 3 2

خروجی نمونه ۲

4

عکس مورد نظر شما پیدا نشد

www.UUupload.ir

در این مثال رنگ‌های مختلف به صورت زیر است:

$\{1\}, \{2\}, \{3\}, \{2, 3\}$

تایپ‌ست بزرگ

- محدودیت زمان: ۰.۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

مصطفی که آدمی کاری است، مسئول تایپ شرکت شده است. چون مصطفی بسیار کاری است آنقدر تایپ کرده است که سرور های شرکت همگی پر شده اند. حالا او قصد دارد که این متن ها را فشرده کند تا دوباره بتواند تایپ کند. برای این کار او به ازای هر کلمه باید عملیات زیر را انجام دهد:

اگر تعدادی حرف متوالی برابر در کلمه بود، به جای آن ها یک بار آن حرف را چاپ و سپس تعداد آن ها را جلوی آن چاپ میکند. مثلا به جای کلمه‌ی WWWSoO، مصطفی W3So2 را چاپ میکند.

دقت کنید که کوچکی و بزرگی حروف اهمیت دارد. همچنین اگر تعداد تکرار یک باشد، عدد یک چاپ نمی‌شود.

به همین ترتیب اگر مصطفی بخواهد از فایل‌های تایپ شده استفاده کند، اول آنها را گسترده و سپس استفاده میکند.

حال به شما تعدادی کلمه داده میشود و به ازای هرکدام شما باید آن کلمه را فشرده و یا گسترده کنید.

ورودی

در سطر اول ورودی n می‌آید که نمایانگر تعداد کلمات است.

در $2 \times n$ سطر بعدی n بار یک عدد و یک کلمه می‌آید که اگر عدد برابر یک باشد یعنی آن کلمه باید فشرده و اگر دو باشد باید گسترده شود. کاراکترهای به کار رفته در این کلمات حروف کوچک و بزرگ انگلیسی می‌باشند. مجموع طول تمام کلمه‌ها از ۱۰۰۰ کاراکتر بیشتر نیست.

$$1 \leq n \leq 100$$

خروجی

خروجی شامل n سطر است که در سطر i م باید فشرده شده و یا گسترده شده ی کلمه ی i م ورودی چاپ شود. مجموع طول خروجی ها از ۱۰۰۰ کمتر است.

مثال

ورودی نمونه

```
2
1
hhpwwwBbTTTTnnP
2
h2pw3BbT4n2P
```

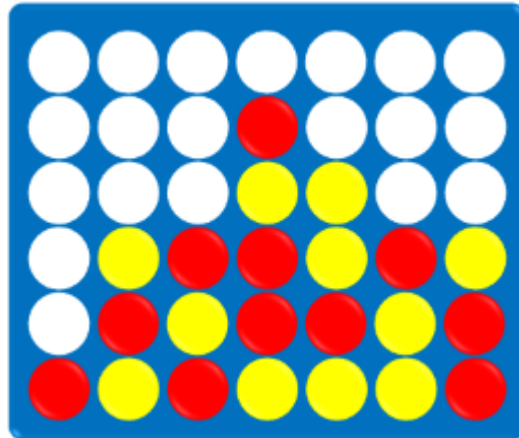
خروجی نمونه

```
h2pw3BbT4n2P
hhpwwwBbTTTTnnP
```

دوزبازی

- محدودیت زمان: ۱۰ ثانیه
- محدودیت حافظه: ۵۱۲ مگابایت

آیا تا به حال بازی دوز ۴ تایی را انجام داده اید؟؟



می‌توانید برای آشنایی بیشتر با بازی دوز از طریق این لینک به صورت آنلاین آن را بازی کنید

باقر و ناصر زمانی که دانشگاه‌ها باز بود هر روز با هم دوز بازی می‌کردند. اما حالا که به خاطر شیوع کرونا دانشگاه‌ها تعطیل شده و نمی‌توانند با هم به رقابت بپردازند، می‌خواهند برنامه‌ای بنویسند تا بتوانند دوز را به صورت کامپیوتری با هم بازی کنند. اما آن‌ها در برنامه‌نویسی ضعیف هستند و برای نوشتن این برنامه از شما که هم اکنون به مهارت خوبی در برنامه‌نویسی پایتون رسیده‌اید کمک می‌خواهند.

حال شما باید برنامه‌ای را بنویسید که حرکات هر بازیکن در بازی دوز را از ورودی دریافت کرده و در خروجی صفحه نهایی بازی و برنده آن را چاپ کند

ورودی

در خط اول ورودی عدد n که تعداد همه حرکات بازی است داده می‌شود. این حرکات از بازیکن قرمز (r) شروع می‌شود. یعنی اگر تعداد 7 حرکت داده شد به این معنی است که بازیکن قرمز 4 حرکت و بازیکن زرد (y) 3 حرکت داشته است. خانه‌ها از چپ به راست با اعداد یک تا هفت شماره‌گذاری شده‌اند. در خط بعدی ورودی n عدد بین 1 تا 7 داده می‌شود که شماره خانه‌ای است که بازیکن صاحب

نوبت مهره اش را درون آن می اندازد (صفحه بازی دوز همیشه دارای ۶ سطر ۷ ستون است). در این برنامه تضمین می شود که هیچ گاه شماره خانه ای که نتوان در آن بازی کرد به عنوان ورودی داده نمی شود.

خروجی

ما تضمین می کنیم که در همه تست های نمونه با به پایان رسیدن حرکات داده شده یکی از بازیکنان برنده می شود. در خروجی برنامه، شما باید نام بازیکن برنده و صفحه نهایی بازی را چاپ کنید.

برای آشنایی بیشتر با فرمت های ورودی و خروجی برنامه به مثال های نمونه دقت کنید

نکته مهم

در پیاده سازی این سوال حتما باید از این سه تابع استفاده شود

۱. تابعی که به عنوان ورودی لیست ۲ بعدی صفحه بازی را گرفته و آن را مشابه آنچه در تست های

نمونه آورده شده چاپ کند

۲. تابعی که به عنوان ورودی لیست ۲ بعدی صفحه بازی و عدد ستون مد نظر بازیکن و نام رنگ

مهره بازیکن را بگیرد. و حرکت مد نظر بازیکن را در صفحه بازی ثبت و صفحه جدید بازی را

برگرداند

۳. تابعی که به عنوان ورودی لیست ۲ بعدی صفحه نهایی و کامل بازی را گرفته و با بررسی صفحه،

رنگ بازیکن برنده را مشخص کرده و برگرداند(طبعا مشخص کردن برنده صرفا از روی تعداد حرکات

انجام شده قابل قبول نیست)

مثال

ورودی نمونه ۱

8

1 2 1 2 1 2 3 2

خروجی نمونه ۱

```

Winner = y
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 y 0 0 0 0 0
r y 0 0 0 0 0
r y 0 0 0 0 0
r y r 0 0 0 0

```

ورودی نمونه ۲

```

11
1 2 2 3 2 3 3 4 4 4 4

```

خروجی نمونه ۲

```

Winner = r
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 r 0 0 0
0 r r y 0 0 0
0 r y r 0 0 0
r y y y 0 0 0

```

ورودی نمونه ۳

```

11
4 5 5 4 6 3 4 3 5 3 3

```

خروجی نمونه ۳

```

Winner = r
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 r 0 0 0 0
0 0 y r r 0 0

```

0 0 y y r 0 0
0 0 y r y r 0

اسفا میل

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

در جمعی n نفر مشغول بازی اسم‌فامیل هستند! در این بازی m موضوع داریم (مثل اسم، فامیلی، غذا، نام شهر و...). و برای هر موضوع نیز تعدادی کلمه‌ی مجاز داریم! همچنین این بازی متشکل از تعدادی مرحله است. در هر مرحله یکی از حروف الفبا انتخاب می‌شود؛ سپس تمام افراد باید به ازای هر موضوع، در کاغذ خود یک کلمه بنویسند که با حرف متناظر آن مرحله شروع می‌شود.

پس از پایان هر مرحله، به هر کس امتیازی می‌رسد (به ازای هر موضوع امتیاز جداگانه‌ای به هر فرد اضافه می‌شود) چنان چه در موضوعی آن شخص کلمه‌ای ننوشته باشد و یا کلمه‌ای نوشته باشد که مجاز نباشد و یا با آن حرف شروع نشود، امتیاز آن موضوع را دریافت نمی‌کند؛ اگر کلمه‌ای که در این موضوع نوشته است، توسط بازیکن دیگری نیز نوشته شده باشد، از این موضوع ۵ امتیاز می‌گیرد و اگر فقط او چنین کلمه‌ای در این موضوع نوشته باشد، ۱۰ امتیاز دریافت خواهد کرد.

مثلاً اگر تنها دو موضوع «اسم» و «فامیل» داشته باشیم، حرفی که برای این مرحله انتخاب می‌شود a باشد و سه بازیکن داشته باشیم که هر یک به شرح زیر کلماتی را در کاغذشان نوشته باشند؛

بازیکن	اسم	فامیل
۱	ali	ahadi
۲	ali	akbari
۳		askari

«اسم»های مجاز: ali, ahmad

«فامیل»های مجاز: akbari, askari

در این بازی، بازیکن یک ۵ امتیاز، بازیکن دو ۱۵ امتیاز و بازیکن سه ۱۰ امتیاز خواهند گرفت.

در ورودی جزییات بازی به شما داده می‌شود و شما باید امتیاز نهایی هر بازیکن را در خروجی چاپ کنید.

ورودی

در خط اول ورودی سه عدد آمده که با space از هم جدا شده‌اند. ابتدا n تعداد بازیکن‌ها، سپس m تعداد موضوعات و سپس k تعداد مرحله‌ها.

در m خط بعد، کلمات مجاز هر موضوع آمده‌اند؛ در خط i ام کلمات مجاز موضوع i ام می‌آیند که با space از هم جدا شده‌اند.

ادامه‌ی ورودی k بخش دارد، بخش i ام نمایان‌گر اطلاعات مرحله‌ی i ام است.

اطلاعات هر مرحله شامل $n + 1$ خط است؛ در خط اول حرفی که برای آن مرحله انتخاب شده می‌آید و سپس در n خط بعد، در هر خط m کلمه می‌آید که -باز هم- با space از هم جدا می‌شوند. کلمه‌ی i ام در خط i ام نمایان‌گر کلمه‌ای است که بازیکن i ام برای موضوع i ام در این مرحله روی کاغذش می‌نویسد. در صورتی که بازیکنی برای موضوعی هیچ کلمه‌ای ننوشته باشد، به جای آن کلمه عبارت "EMPTY" در ورودی می‌آید.

لازم به ذکر است که هر کلمه متشکل از حداکثر ۱۰ کاراکتر است و حرف متناظر هر مرحله از بازی و تمام کاراکترهای کلمات از بین حروف کوچک الفبای انگلیسی انتخاب می‌شوند. همچنین تعداد مراحل بازی حداکثر ۲۶ و تعداد بازیکن‌ها و موضوعات حداکثر ۵۰ عدد خواهد بود. تعداد کلمات مجاز به ازای هر موضوع نیز از ۱۰۰ کلمه بیشتر نخواهد شد.

خروجی

در تنها خط خروجی n عدد چاپ کنید که عدد i ام امتیاز نهایی بازیکن i ام است. اعداد باید با space از هم جدا شده باشند.

مثال

ورودی نمونه ۱

4 3 2

karim rahim asghar ahmad zahra zeynab

zakeri karimi rahimi akbari ahmadi zahiri

zebra zaloo moorcheh zarrafeh soosk palang ankaboot rasoo

z

EMPTY zahiri EMPTY

zeynab zakeri zebra

karim zahiri zebra

zohreh EMPTY zaloo

a

ahmad ahmadi ankaboot

asghar akbari ankabtoo

rahim rahimi rasoo

EMPTY EMPTY asb

خروجی نمونه ۱

35 45 10 10

در مرحله ی اوّل امتیازاتی که افراد می گیرند به این شکل است:

بازیکن ۱: ۵ امتیاز

بازیکن ۲: ۲۵ امتیاز

بازیکن ۳: ۱۰ امتیاز

بازیکن ۴: ۱۰ امتیاز

در مرحله ی دوم نیز امتیازات به این شکل خواهند بود:

بازیکن ۱: ۳۰ امتیاز

بازیکن ۲: ۲۰ امتیاز

بازیکن ۳: ۰ امتیاز

بازیکن ۴: ۰ امتیاز

بایتون

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

زبان برنامه‌نویسی *Bython* که به تازگی عرضه شده است، فقط دستورات زیر را دارد:

نام دستور	فرم کلی دستور	توضیح دستور
<i>Assignment</i>	$A = expression$	مقدار <i>expression</i> را در متغیر <i>A</i> ذخیره می‌کند.
<i>Input</i>	<i>voroodi()</i>	یک عدد از ورودی می‌خواند و آن را بر می‌گرداند.
<i>Print</i>	<i>(khoorooji(List</i>	مقادیر موجود در <i>List</i> را به ترتیب چاپ می‌کند.
<i>If Statement</i>	$agar\ condition : A = expression$	در صورتی برقراری <i>condition</i> ، مقدار <i>expression</i> در متغیر <i>A</i> ذخیره می‌شود.

که در جدول بالا مقادیر *expression* ، *List* ، *condition* و متغیر به این صورت تعریف می‌شوند:

• تعریف *expression*:

- هر عدد ثابتی یک *expression* است. برای مثال اعداد ۲ و ۵، خود یک *expression* هستند.
- مقدار *voroodi()* به تنهایی یک *expression* است.
- هر عبارتی به شکل $A + B$ و یا $A - B$ که در آن، A و B می‌توانند متغیر و یا عدد ثابت باشند، یک *expression* است. برای مثال عبارتهای $A + D$ و همچنین $VAR + 5$ که در آن A ، D و VAR متغیر هستند، یک *expression* اند.
- باقی عبارات *expression* نیستند. برای مثال مقادیر $A + B + C$ و یا $A \times B$ یک *expression* نیستند.

• تعریف *List*:

- عبارت A که در آن A متغیر و یا عدد ثابت است، یک لیست است.
- اگر A و B دو *List* باشند، عبارت A, B نیز یک *List* است.

- برای مثال اگر A و B و D متغیر باشند، عبارات 3 و A و 7 و A و همچنین $D, B, 1$ یک *List* هستند.
- تعریف *condition*:
 - عبارت $A == B$ که در آن A و B متغیر و یا عدد ثابت هستند یک *condition* است.
 - در صورتی که مقادیر A و B برابر باشند، *condition* برقرار است و در غیر این صورت *condition* برقرار نیست.
 - برای مثال $3 == 3$ یک *condition* است که برقرار است و عبارت $5 == A$ که در آن مقدار A هنگام اجرا ۶ است، یک *condition* است که برقرار نیست.
- تعریف متغیر:
 - هر رشته‌ای از ترکیب اعداد ۰ تا ۹ و همچنین حروف کوچک و بزرگ الفبای انگلیسی به طول حداکثر ۱۰ که با عدد شروع نشود، یک متغیر است.
 - برای مثال عبارات A ، $VAR2$ و $ABC123de$ یک متغیراند در صورتی که عبارات $3VAR$ ، VAR_3 و $VARIABLE123$ متغیر نیستند.

حال به شما دنباله‌ای از دستورات زبان *Bython* و همچنین دنباله‌ای از مقادیر ورودی داده شده است و از شما خواسته شده است تا خروجی را گزارش کنید. و پس از آن، تعداد متغیرهایی که حداقل یک بار مقداردهی شده‌اند را نیز چاپ کنید.

ورودی

ورودی شامل دو بخش است که در بخش اول دستورات برنامه در چند خط آمده است و پس از آن در یک خط رشته‌ی ----- آمده است سپس ورودی‌های برنامه، هر کدام در یک خط آمده است.

تعداد خط‌های دستورات از ۵۰ بیشتر نخواهد بود.

تضمین می‌شوند تعداد ورودی‌های برنامه برابر با تعداد دستورهای *Input* است و همچنین هر متغیر قبل از استفاده مقداردهی می‌شود.

تمام اعداد داده شده در دستورات، اعداد طبیعی و مثبت خواهند بود، اما ممکن است در هنگام اجرای دستورات، مقدار متغیری منفی شود.

خروجی

در خروجی، به ازای هر دستور *Print* خروجی آن را در یک خط چاپ کنید. سپس در یک خط، تعداد متغیرهایی که حداقل یکبار مقداردهی شده‌اند را خروجی دهید.

تضمین می‌شود حداقل یک دستور *Print* در دستورات برنامه وجود دارد.

زیرمسئله‌ها

محدودیت‌ها	نمره	زیرمسئله
فقط دستور Assignment و Print موجود است و در دستور Print تنها یک ورودی داریم.	۱۰	۱
دستور If Statement نداریم.	۳۰	۲
بدون محدودیت	۶۰	۳

ورودی نمونه ۱

```
A = 3
B = voroodi()
C = A + B
khoorooji(10)
khoorooji(C, 2)
-----
3
```

خروجی نمونه ۱

```
10
6 2
3
```

مقدار متغیر *C* پس از اجرای دستورات برابر ۶ خواهد بود و تعداد متغیرهایی که حداقل یکبار مقداردهی شده‌اند برابر با ۳ است. (متغیرهای *A* و *B* و *C*)

ورودی نمونه ۲

```
A = voroodi()
agar A == 2 : A = A + 2
khoorooji(A)
-----
2
```

خروجی نمونه ۲

```
4
1
```

مقدار متغیر A پس از اجرای دستورات برابر ۴ خواهد بود و تعداد متغیرهایی که حداقل یکبار مقداردهی شده‌اند برابر با ۱ است. (متغیر A)

ورودی نمونه ۳

```
agar 2 == 2 : A = 3
agar 3 == 4 : B = 1
khoorooji(A)
-----
```

خروجی نمونه ۳

```
3
1
```

مقدار متغیر A پس از اجرای دستورات برابر ۳ خواهد بود و تعداد متغیرهایی که حداقل یکبار مقداردهی شده‌اند برابر با ۱ است. (متغیر A)

ایزی لیگ

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

علی به تازگی با برنامه نویسی آشنا شده است و برای همین ذوق خاصی دارد. از این رو می‌خواهد سیستمی طراحی کند که یک لیگ فوتبال را شبیه سازی می‌کند.

در این سیستم، دستورات مختلفی وجود دارند که به شرح زیر می‌باشند:

ساختن بازیکن جدید

new player name price speed finishing defence

این دستور، بازیکنی با نام name، قیمت price، سرعت speed، قدرت تمام‌کنندگی finishing و دفاع defence می‌سازد.

توجه کنید که name از جنس رشته، و باقی پارامترها از جنس عدد می‌باشند.

هر بازیکن، یک شناسه‌ی عددی دارد که با استفاده از آن می‌توان به آن بازیکن دسترسی داشت، اولین بازیکنی که ساخته می‌شود شناسه‌ی عددی ۱، دومین بازیکنی که ساخته می‌شود شناسه‌ی عددی ۲ و i -امین بازیکنی که ساخته می‌شود شناسه‌ی عددی i دارد.

ساختن تیم جدید

new team name money

با استفاده از این دستور، می‌توانید یک تیم جدید با نام name و پول money بسازید. در صورتی که از قبل تیمی با اسم name وجود داشته باشد، تیم جدیدی ساخته نمی‌شود.

همچنین name از جنس رشته و money از جنس عدد می‌باشد.

هر تیم، یک شناسه‌ی عددی دارد که با استفاده از آن می‌توان به آن تیم دسترسی داشت، اولین تیمی که ساخته می‌شود شناسه‌ی عددی ۱، دومین تیمی که ساخته می‌شود شناسه‌ی عددی ۲ و i -امین تیمی که ساخته می‌شود شناسه‌ی عددی i دارد.

خرید بازیکن توسط تیم

buy playerID teamID

با استفاده از این دستور، تیم با شناسه‌ی عددی teamID بازیکن با شناسه‌ی عددی playerID را می‌خرد.

همچنین همانطور که در بالا گفته شد، هر بازیکن یک قیمت (price) دارد و هر تیمی مقداری پول money دارد و هنگام خرید یک بازیکن باید تیم مورد نظر پول کافی را برای خرید بازیکن داشته باشد و در صورتی که خرید موفقیت‌آمیز باشد، از پول تیم، به میزان قیمت بازیکن مورد نظر کم می‌شود.

توجه کنید که در امر خرید بازیکن ممکن هست حالت‌های مختلفی پیش بیاید:

- در صورتی که بازیکنی با شناسه‌ی عددی playerID وجود نداشته باشد، باید عبارت player with the id playerID doesnt exist را چاپ کنید.
- در صورتی که تیمی با شناسه‌ی عددی teamID وجود نداشته باشد، باید عبارت team with the id teamID doesnt exist را چاپ کنید.
- در صورتی که تیم مورد نظر، پول کافی برای خرید بازیکن را نداشته باشد باید عبارت the team cant afford to buy this player را چاپ کنید.
- در صورتی که بازیکن مورد نظر، در تیمی عضو باشد، باید عبارت player already has a team را چاپ کنید.

در صورتی که هیچ‌کدام از حالت‌های بالا پیش نیایند، با موفقیت بازیکن مورد نظر خریداری می‌شود و باید عبارت player added to the team succesfully را چاپ کنید.

توجه کنید که ممکن است چندین مورد از حالت‌های بالا پیش بیایند، در این صورت شما باید اولین حالتی که پیش می‌آید را در نظر گرفته و خروجی مورد نظر را چاپ کنید و از بقیه حالت‌ها صرف نظر کنید.

فروش بازیکن

`sell playerID teamID`

برای فروش بازیکن می‌توان از دستور بالا استفاده کرد. با استفاده از این دستور، تیم با شناسه‌ی عددی `teamID` بازیکن با شناسه‌ی عددی `playerID` را به فروش می‌رساند.

همچنین موقع فروش بازیکن ممکن است حالت‌های مختلفی پیش بیایند:

- در صورتی که تیمی با شناسه‌ی عددی `teamID` وجود نداشته باشد، باید عبارت `team doesnt exist` را چاپ کنید.
- در صورتی که تیم مورد نظر وجود داشته باشد اما بازیکنی با شناسه‌ی عددی `playerID` در آن تیم وجود نداشته باشد، باید عبارت `team doesnt have this player` را چاپ کنید.

در صورتی که حالت‌های بالا پیش نیایند، یعنی با موفقیت بازیکن مورد نظر فروخته شده است و این بازیکن دیگر به تیمی تعلق ندارد (ممکن است در آینده توسط تیمی خریداری شود). در این حالت به اندازه‌ی قیمت بازیکن فروخته شده، به پول تیم مورد نظر افزوده می‌شود. همچنین در این حالت باید عبارت `player sold succesfully` را چاپ کنید.

انجام بازی بین دو تیم

`match teamID1 teamID2`

با استفاده از این دستور، دو تیم با شناسه‌های عددی `teamID1` و `teamID2` با یکدیگر بازی می‌کنند.

همچنین ممکن است در اجرای این دستور، حالت‌های مختلفی پیش بیاید:

- اگر حداقل یکی از دو تیم وجود نداشته باشد، باید عبارت `team doesnt exist` را چاپ کنید.
- اگر دو تیم مورد نظر وجود داشتند، اما حداقل یکی از آن‌ها تعداد بازیکن‌هایش کمتر از ۱۱ نفر باشد، بازی انجام نمی‌شود و باید عبارت `the game can not be held due to loss of the players` را چاپ کنید.

در صورتی که هیچکدام از مشکلات بالا پیش نیاید، بازی با موفقیت انجام می‌شود.

در ابتدا برای توصیف روند بازی، نیاز به تعریف پارامتر قدرت برای هر تیم داریم.

قدرت تیم اول (تیم با شناسه‌ی عددی `teamID1`) برابر است با مجموع سرعت و تمام‌کنندگی ۱۱ بازیکن اول آن تیم.

همچنین قدرت تیم دوم (تیم با شناسه‌ی عددی `teamID2`) برابر است با مجموع سرعت و دفاع ۱۱ بازیکن اول آن تیم.

توجه کنید که ترتیب بازیکن‌ها در یک تیم، همان ترتیب زمانی اضافه شدن آن‌ها به تیم می‌باشد.

در صورتی که قدرت دو تیم باهم برابر نباشد، تیمی برنده‌ی بازی می‌شود که قدرت بیشتری دارد و به تعداد بردهای تیم برنده یک واحد اضافه می‌شود و به تعداد باخت‌های تیم بازنده نیز یک واحد اضافه می‌شود و به تیم برنده، به عنوان جایزه ۱۰۰۰ واحد پول داده می‌شود. در صورتی که قدرت دو تیم برابر باشد، بازی مساوی می‌شود و تعداد مساوی‌های هر دو تیم یک واحد افزایش پیدا می‌کند.

نمایش جدول تیم‌ها

rank

با استفاده از دستور بالا، باید جدول رتبه‌بندی تیم‌ها را چاپ کنید.

شما باید تیم‌ها را بر اساس تعداد بردها (هر تیمی که برد بیشتری داشته باشد رتبه بهتری دارد)، و در صورت برابری تعداد بردها، بر اساس تعداد باخت‌های کمتر مرتب کنید (اگر دو تیم بردهای برابری داشته باشند، تیم با باخت کمتر رتبه بهتری می‌گیرد) و به صورت زیر نام آن‌ها را چاپ کنید:

```
1. teamName[0]
2. teamName[1]
...
```

که در آن `teamName[0]` تیم با رتبه‌ی ۱، `teamName[1]` تیم با رتبه‌ی ۲ و... می‌باشند.

توجه کنید که تعداد مساوی‌ها مهم نمی‌باشد و همچنین اگر دو تیم تعداد باخت‌ها و بردهایشان برابر بود، تیمی که شناسه‌ی عددی کمتری دارد رتبه‌ی بهتری می‌گیرد.

پایان لیگ

end

در انتها، پس از تمام شدن لیگ، عبارت بالا در خروجی داده می‌شود و به این معنی است که ورودی‌ها تمام شده است و برنامه شما نیز پایان می‌یابد.

ورودی

ورودی شامل چندین خط می‌باشد که در هر خط یکی از دستوراتی که در صورت سوال گفته شد، داده می‌شود.

همچنین در خط آخر ورودی، دستور end داده می‌شود.

توجه کنید که ممکن است در ورودی داده شده، بعضی از خط‌ها خالی باشند و هیچ دستوری در آن‌ها نباشد.

خروجی

پس از اجرای برخی از دستورها، باید خروجی‌هایی را چاپ کنید که در توضیحات هر دستور در صورت نیاز گفته شده است. توجه کنید که هر خروجی باید در یک خط چاپ شود و از چاپ تمامی خروجی‌ها در یک خط پرهیز کنید.

مثال

ورودی نمونه ۱

```
new player ArshiA 1 2 3 4
new player sadegh 1 2 3 4
new player haniye 1 2 3 4
new player fateme 1 2 3 4
new player qorban 1 2 3 4
new player abbasi 1 2 3 4
new player alikmr 1 2 3 4
new player sadegh 1 2 3 4
```

```
new player hamide 1 2 3 4
new player karane 1 2 3 4
new player vanaki 1 2 3 4
```

```
new player Ars 1 100 100 100
new player sad 1 100 100 100
new player han 1 100 100 100
new player fat 1 100 100 100
new player qor 1 100 100 100
new player abb 1 100 100 100
new player ali 1 100 100 100
new player sad 1 100 100 100
new player ham 1 100 100 100
new player kar 1 100 100 100
new player van 1 100 100 100
```

```
new team t2 100
new team t1 100
buy 12 1
buy 13 1
buy 14 1
buy 15 1
buy 16 1
buy 17 1
buy 18 1
buy 19 1
buy 20 1
buy 21 1
buy 22 1
```

```
buy 1 2
buy 2 2
buy 3 2
buy 4 2
buy 5 2
buy 6 2
buy 7 2
buy 8 2
buy 9 2
buy 10 2
buy 11 2
match 1 2
```

```
rank  
end
```

خروجی نمونه ۱

```
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
player added to the team succesfully  
1. t2  
2. t1
```