

به نام خدا



درس برنامه سازی پیشرفته

تمرین ۰

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۲ - ۰۱

استاد:

محمد امین فضلی

مهلت ارسال:

۲۶ اسفند ساعت ۲۳:۵۹

مسئول تمرین ها:

هیربد بهنام، پرهام چاوشیان و میلاد سعادت

مسئول تمرین ۰:

محسن قاسمی

طراحان تمرین ۰:

امیرحسین حاج محمد رضایی

علی شاه علی

علی صادقی

مهدی لطیفیان

امیرحسین کوچکیان

فهرست مطالب

	سوالات
۲	سوال ۱. رمز حلقه
۲	سوال ۲. حلقه آینه‌ای
۴	سوال ۳. تقابل برومیر و فرودو
۶	سوال ۴. ساخت حلقه تاریکی
۱۱	سوال ۵. فروشگاه آنلاین
۱۷	

سوالات

سوال ۱. رمز حلقه



هنگامی که گاندالف متوجه شد که حلقه‌ای که در دست بیلبو بگینز است در واقع همان حلقه تاریکی است که توسط سائورون ساخته شده است، برای نابود کردن حلقه شروع به مطالعه درباره آن کرد. در حین مطالعاتش او متوجه شد که هنگامی که حلقه در آتش قرار می‌گیرد، نوشته‌هایی بر آن ظاهر خواهند شد و او باید رمز بر این متن را بگشاید تا راهی برای نابود کردن سائورون بیابد. دستورالعمل رمزگشایی متن به صورت زیر است: بر حلقه کلماتی وجود دارد که بوسیله - از یکدیگر جدا شده‌اند. دو یا چندین کلمه را هم‌ریشه می‌خوانیم اگر مجموعه حروف آن کلمات با هم یکسان باشند، به عنوان مثال سه کلمه ate، eat و tea با یکدیگر هم‌ریشه هستند. گاندالف باید کلمات هم‌ریشه را تشخیص دهد و در گروه‌های جداگانه قرار دهد. از آنجا که او از شدت مطالعه زیاد خسته شده است، از شما یاری می‌طلبد تا با نوشتن یک برنامه، به او در رمزگشایی حلقه کمک کنید.

ورودی

و رودی تنها یک خط است که آن یک رشته از کلمات است که بوسیله کاراکتر - از یکدیگر جدا شده‌اند.

$$1 \leq m \leq 200$$

$$0 \leq n \leq 50$$

$$1 \leq k \leq 10$$

تعداد کلمات برابر با m ، طول کلمات در یک رشته برابر با n و تعداد گروه‌بندی کلمات هم ریشه نیز برابر با k می‌باشد.

خروجی

در k خط خروجی، کلمات هم‌ریشه در یک خط چاپ شوند. هر کلمه بین دو علامت ستاره همانند در مثال‌ها چاپ شود. دقت کنید که برای ترتیب خروجی دادن کلمات و گروه‌بندی‌ها، به همان ترتیب ظاهر شدن در رشته، آنها را چاپ کنید. اگر کلمه‌ای در رشته چندین بار تکرار شده بود، آن را در خروجی فقط یکبار چاپ کنید. و برای ترتیب آن، فقط دفعه اول ظاهر شدن آن را در نظر بگیرید.

ورودی نمونه ۱

```
bat-eat-tea-tan-ate-nat-tea
```

خروجی نمونه ۱

```
*bat*
*eat* *tea* *ate*
*tan* *nat*
```

در اینجا به ترتیب وقوع کلمات در رشته دقت کنید که به همان ترتیب در خروجی چاپ شده‌اند.

سوال ۲. حلقه آینه‌ای

علی که یکی از طرفدارهای دنیای ارباب حلقه‌ها عه، نمیخواست با داستان‌های بیمزه اون مجموعه شاهکار رو خراب کنه. بنابراین سوالش رو به صورت خلاصه در ادامه آورده و از شما میخواد اونو براش حل کنید.

دنباله a به طول n به ما داده شده است. می‌خواهیم این دنباله را به یک دنباله پالیندروم تبدیل کنیم. برای این کار میتوانیم تعدادی از اعضای a که مقدار برابری دارند را از دنباله حذف کرده و اعضای باقیمانده را به هم بچسبانیم تا دنباله جدید حاصل شود. حال مشخص کنید آیا اینکار امکان‌پذیر است یا خیر. می‌گوییم یک دنباله پالیندروم یا آینه‌ای است اگر و تنها اگر از هر دو سمت به طور یکسان خوانده شود.

ورودی

در خط اول ورودی عدد T می‌آید که تعداد تست‌کیس‌ها را نشان می‌دهد. در ادامه T سناریو آمده که باید برای هر کدام مساله را حل کنید. در خط اول هر سناریو عدد n و در خط بعد دنباله a به طول n از اعداد طبیعی آمده‌است. تضمین میشود مجموع n به ازای تمام سناریوها حداکثر ۲۰۰۰۰۰ است.

$$1 \leq T \leq 10000$$

$$1 \leq n \leq 200000$$

$$1 \leq a_i \leq n$$

خروجی

به ازای هر سناریو در یک خط جدا یکی از دو کلمه YES یا NO را چاپ کنید. (به بزرگ بودن حروف دقت کنید.)

ورودی نمونه ۱

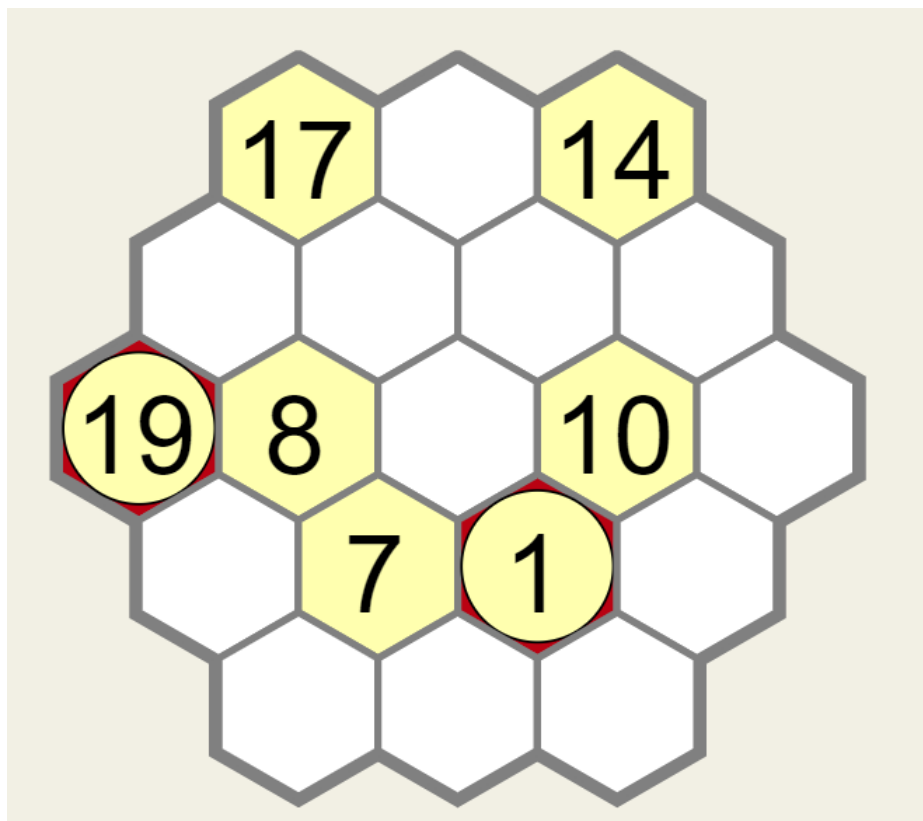
4
1
1
2
1 2
3
1 2 3
5
1 4 4 1 4

خروجی نمونه ۱

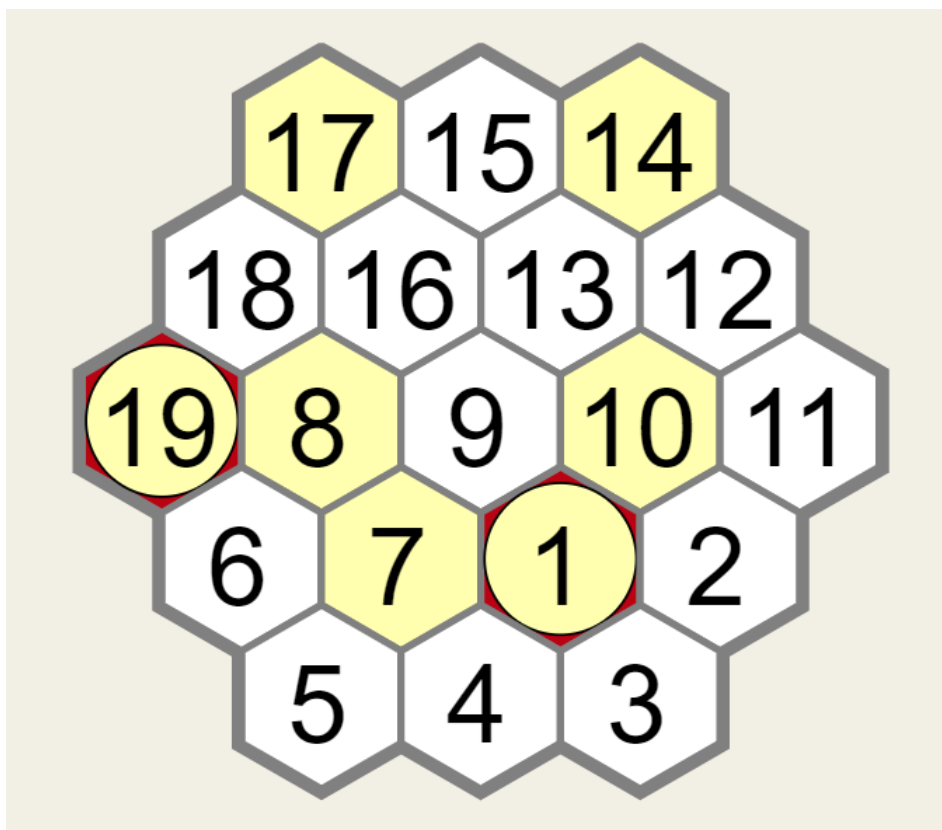
YES
YES
NO
YES

سوال ۳. تقابل برومیر و فرودو

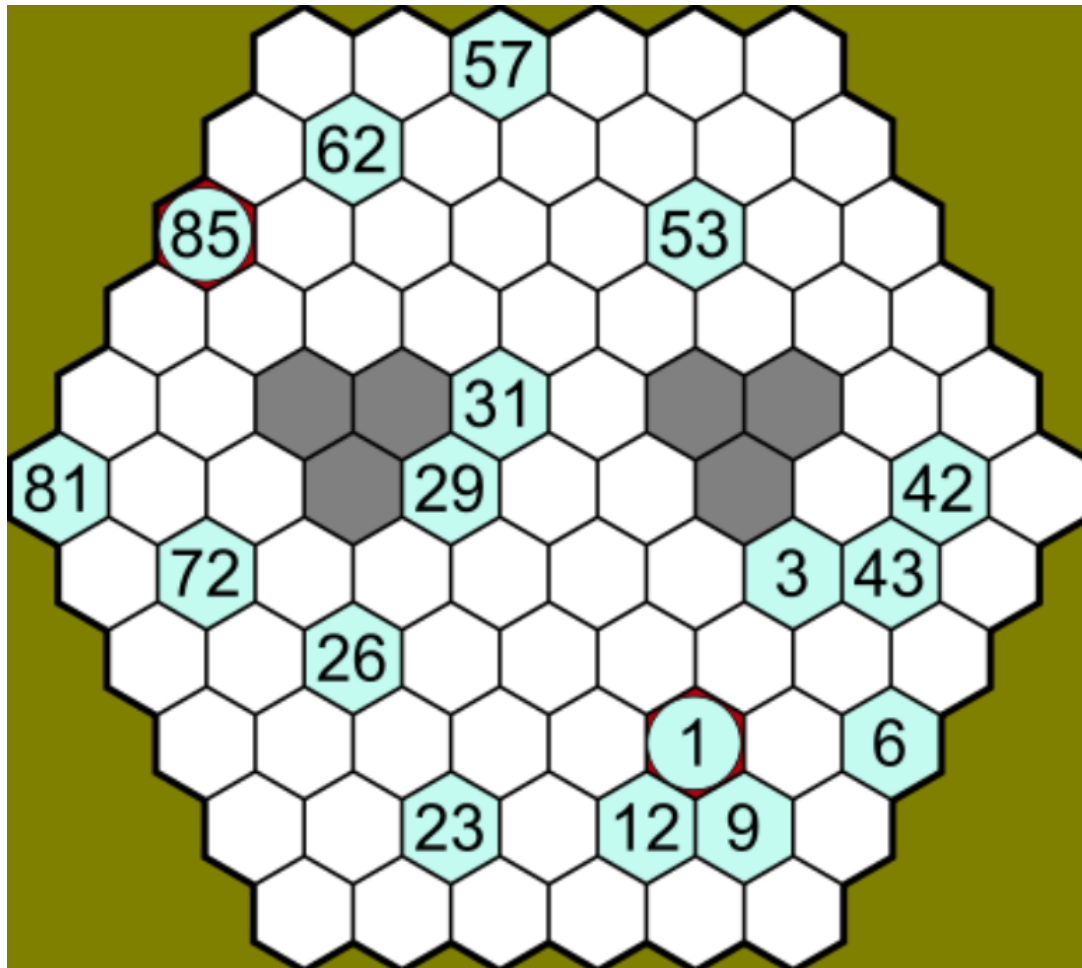
برومیر قصد گرفتن حلقه از فرودو را دارد و فرودو برای اینکه برومیر را به زحمت بیندازد، بازی ای شبیه حلقه برای او ساخت تا در صورت برنده شدن در آن، حلقه را به او بدهد. فرودو رو به برومیر کرد و گفت که تو باید با اعدادی که می‌خواهم حلقه ای بسازی که به همدیگر متصل باشند. یعنی اعداد آن از ۱ تا n به صورت متوالی با هم مجاور باشند و چون فرودو عاشق اشکال شش ضلعی بود، صفحه اش را به ۶ ضلعی‌ها افراز کرد و شش ضلعی جدیدی پدید آورد و گفت که هر خانه ۶ ضلعی، فقط با ۶ خانه ای که ضلع مشترک دارد مجاور است. در ادامه توضیحات کاملتری که فرودو برای برومیر داده را آورده ایم.



این تصویر شکل یک شش ضلعی با ضلع ۳ است که در مجموع ۱۹ خانه دارد. همان‌طور که مشخص است بعضی از اعداد را فرودو از اول در جایی قرار داده و برومیر نمی‌تواند آن‌ها را جابه‌جا کند. حال برومیر باید دیگر اعداد را به گونه‌ای در خانه‌های خالی شکل قرار دهد که هر دو عدد متوالی با هم همسایه باشند. که جواب این نمونه را ادامه می‌بینید.



فرودو برای سخت تر کردن بازی، خانه های بلاک شده برای برومیر قرار داد تا نتواند به راحتی حلقه را از او بگیرد. برومیر در این خانه ها هیچ عددی نمیتواند قرار دهد و این خانه هارا با 1- نشان میدهد. در واقع خانه های خاکستری رنگ در شکل زیر، نمونه ای از این سختگیری فرودو برای برومیر هستند.



برومیر که به شدت در تلاش برای بدست آوردن حلقه است، از شما می‌خواهد تا با زدن کدی به زبان جاوا، جدول فرودو را برایش حل کرده و حلقه را از چنگ او دراورید.

ورودی

در خط اول عدد n آمده است که طول ضلع شش ضلعی است.

$$1 \leq n \leq 6$$

در $2n-1$ خط بعدی اعداد داخل جدول آمده‌اند. تعداد در هر خط به این‌گونه است که خط اول n عدد دارد و تا خط n ام در هر خط تعداد اعداد یکی زیاد می‌شود تا تعداد ورودی‌ها در خط n ام به $2n-1$ برسد. ادامه خطوط در هر خط تعداد اعداد یکی کم می‌شود تا دوباره در خط $2n-1$ ام تعداد ورودی دوباره به n می‌رسد.

عدد 1- در جدول نشانه خانه‌های بلوکه شده هستند و نمی‌توان در آن خانه عددی قرار داد. عدد ۰ به معنای خانه‌های خالی و اعداد بیش‌تر از ۰ یعنی همان عدد در آن خانه قرار دارد و نمی‌توان آن را جابه‌جا کرد. تضمین می‌شود که عدد ۱ و مقدار ماکسیمم در جدول وجود دارد.

به طور مثال برای $n=3$ اگر هر عدد را با حرف q نشان دهیم ساختار ورودی به صورت زیر است.

```
3
q q q
q q q q
q q q q q
q q q q
q q q
```

خروجی

در خروجی جدول حل شده را مشابه ورودی چاپ کنید.

نکته: از آن‌جا که ممکن است برای یک جدول، چند جواب صحیح یافت شود، شما می‌توانید هر یک از این جواب‌ها را به دلخواه در خروجی چاپ کنید و این مورد در نمره داوری تأثیر نخواهد داشت.

ورودی نمونه ۱

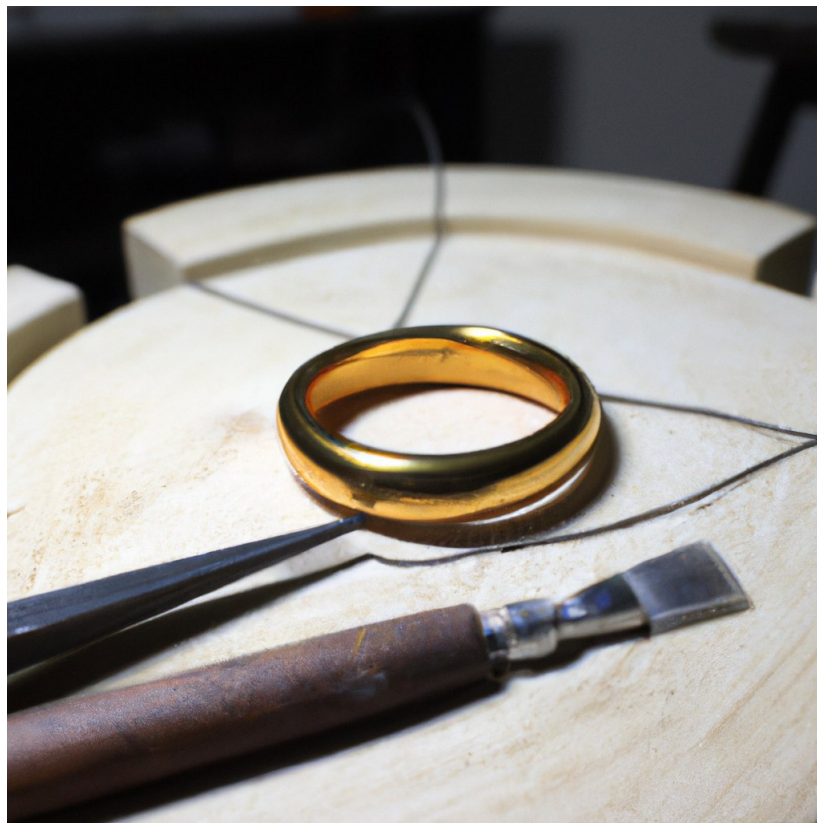
```
3
17 0 14
0 0 0 0
19 8 0 10 0
0 7 1 0
0 0 0
```

خروجی نمونه ۱

17 15 14
18 16 13 12
19 8 9 10 11
6 7 1 2
5 4 3

سوال ۴. ساخت حلقه تاریکی

سائورون به صورت مخفی تغییراتی روی داکيومنت‌های ساخت حلقه‌ها انجام داد تا حلقه تاریکی را طراحی کند و پس از ساخت آن به سرعت از middle earth خارج شد. ارلوند پس از پی بردن به موضوع سعی کرد تا تغییراتی که سائورون روی داکيومنت‌ها بوجود آورده را بازیابی کند تا بفهمد حلقه تاریکی چه توانایی‌هایی دارد. او به کمک سلبریمبور موفق میشود تغییراتی که سائورون روی داکيومنت‌ها ایجاد کرده به صورت دنباله‌ای از دستورات برای هر داکيومنت بازیابی کند اما دستورات طولانی و پشت سر هم بدون نظم خاصی آمده‌اند و الوند توان فهم آن‌ها را ندارد. بنابراین از شما کمک میخواهد تا هر دستور را به به اجزای تشکیل دهنده آن بشکنید تا بتواند متوجه شود سائورون چگونه حلقه تاریکی را ساخته است.



تصویر بالا توسط ChatGPT تولید شده است.

در مورد bash script و دستورات آن می‌توانید در **این لینک** بیشتر مطالعه کنید.
هر دستور می‌تواند از ۵ بخش، command, subcommand, options, arguments, flags تشکیل شده باشد که شما باید آن‌ها را از هم جدا کنید.

command

اولین بخش هر دستور، command است که از یک کلمه تشکیل شده و فضای دستور را به صورت گروهی از دستورها یا یک تک دستور مشخص می‌کند.

```
git commit -m "magnified ring power" -break;
```

برای مثال در دستور بالا git بخش command دستور را تشکیل می‌دهد.

subcommand

بخش subcommand گاهی اوقات پس از بخش command می‌آید و نوع دستور را به صورت یکتا مشخص می‌کند.

```
git commit -m "magnified ring power" -break;
```

برای مثال در دستور بالا commit بخش subcommand دستور را تشکیل می‌دهد.

option

بخش option با یک خط تیره (dash) و یک حرف پس از آن مشخص می‌شود و وظیفه‌اش تغییر دادن رفتار دستور است. option‌ها گاهی با یک argument که بعد از آن می‌آید همراه هستند.

```
git commit -m "commit message" -break;
```

برای مثال در دستور بالا -m بخش option دستور را تشکیل می‌دهد و "commit message" ورودی است که به دنبال آن آمده است. ممکن است چند option پشت سر هم تنها با یک خط تیره ابتدایشان مشخص شوند. در این صورت ورودی بعد از آن (در صورت وجود) مربوط به آخرین option می‌باشد.

```
ls -la;
```

برای مثال در دستور بالا l و a دو option دستور داده شده هستند.

flag

بخش flag با دو خط تیره (dash) که ابتدای آن آمده است مشخص می‌شود و وظیفه‌اش فعال یا غیرفعال کردن یک ویژگی در دستور مورد نظر است.

```
git commit -m "magnified ring power" -break;
```

برای مثال در دستور بالا -break بخش flag را تشکیل می‌دهد.

argument

بخش arguments شامل ورودی‌هایی است که ممکن است بعد از یک option یا به طور مستقل برای command داده شود.

نکته: در صورتی که argument داخل double quotation باشد، می‌تواند شامل هر حرف و علامتی باشد. اما در غیر این صورت تنها شامل حروف، اعداد و نقطه می‌تواند

باشد.

نکته: argument هایی که بلافاصله بعد از بخش command می‌آیند، حتما داخل double quotation قرار می‌گیرند.

```
echo "ring construction will be completed in 2 hours";
```

برای مثال در دستو بالا "ring construction will be completed in 2 hours" ورودی مستقل دستور echo است.

pipe

می‌دانیم که دستورات ممکن است به کمک علامت '|' به هم متصل شوند (که اصطلاحاً به آن پایپ گفته می‌شود). به این صورت که خروجی دستور قبل به عنوان ورودی دستور بعدی استفاده شود. در این صورت نام دستوراتی که قبل از دستور فعلی آمده و به آن متصل شده‌اند باید به ترتیب چاپ شوند تا الیروند بتواند ورودی دقیق هر دستور را به صورت مستقل شناسایی کند.

save & print

اگر در انتهای یک دستور علامت '>' بیاید، در ادامه نام یک فایل داده می‌شود و خروجی ساخته شده علاوه بر چاپ شدن باید در فایل با نام داده شده ذخیره شود. در صورتی که علامت '>>' بیاید خروجی ساخته شده باید به محتویات فایلی که نامش در ادامه می‌آید append شود.

در ادامه با آمدن دستور cat محتویات فایل، پس از مشخص کردن بخش‌های مختلف دستور در خروجی، چاپ می‌شود.

نکته: نام فایل حتما داخل double quotation می‌آید.

ورودی

در ورودی دستورات به صورت پی در پی داده شده و با دستور exit پایان می‌یابند (نیازی

به تجزیه دستور exit نیست).
دستورات و همچنین عبارات موجود در هر دستور ممکن است با یک یا چند فاصله، تب یا اینتر از هم جدا شده باشند.

خروجی

به ازای هر دستور داده شده در ورودی، در خط اول شماره دستور، در خط دوم command مربوطه، در خطوط بعدی به ترتیب subcommand مربوطه، option ها، flag ها و - argument ها در صورت وجود هر کدام در یک خط می‌آیند. همچنین argument های مربوط به option ها در روبروی option نیز با یک علامت تساوی ظاهر می‌شوند. در صورتی که چند option یا flag یا argument در یک دستور وجود داشته باشد، اولیت چاپ با آن است که زودتر در دستور اصلی آمده باشد. در خطوط آخر مربوط به هر دستور، دستوراتی که خروجی آن‌های ورودی دستور مذکور را تغییر داده به ترتیب از ابتدا به انتها چاپ می‌شوند. اطلاعات هر دستور با یک خط خالی از بقیه دستورات جدا می‌شود.

ورودی نمونه ۱

```
git commit -m "commit message" --break; echo "a message to print"; ls -la | grep .txt;
exit;
```

خروجی نمونه ۱

```
command 1
command: git
subcommand: commit
option 1: m = "commit message"
flag 1: break
argument 1: "commit message"

command 2
command: echo
argument 1: "a message to print"
```


command 3

command: ls

option 1: l

option 2: a

command 4

command: grep

subcommand: .txt

input 1: command 3

سوال ۵. فروشگاه آنلاین

گالادریل پس از شکست هایش فهمید بهتر است از قدرت هایش برای جیب خودش استفاده کند. بنابراین تصمیم گرفت که یک فروشگاه آنلاین درست کند. او می خواهد کمی از قدرتش سواستفاده کند و با خواندن ذهن ملت علایق آنها را شناسایی کند تا همان محصولات را به فروشگاه اضافه کند. علارغم توانایی های بسیارش، او در کد زنی مهارتی ندارد و بنابراین از شما کمک میخواهد. در این سوال قصد داریم تا یک فروشگاه آنلاین مطابق میل گالادریل پیاده سازی کنیم ، فروشگاهیه که از مشتری ، تعدادی دسته کالا و تعدادی کالا تشکیل شده است. در ادامه به توضیح دستورات می پردازیم.

دستورات

شروع برنامه

```
start
```

با این دستور فروشگاه شروع به کار میکند. تا قبل آن هیچ دستوری پذیرفته نیست.

افزودن مشتری

```
add customer [id] [name]
```

این دستور یک مشتری با یک id یکتا و یک اسم اضافه می کند. در صورتی که id تکراری باشد ، باید خطای زیر نمایش داده شود :

The id already exists

افزودن category

add category [name]

خطاها

- اگر در غیر از انتهای اسم category عدد بیاید، آنگاه این دستور یک دستور نامعتبر است.
- اگر در اسم از کارکتری غیر از حروف و اعداد استفاده شود باید خطای زیر نمایش داده شود :

Category format is invalid

- اگر نام کتگوری تکراری بود باید خطای زیر نمایش داده شود :

The category already exists

افزودن کالا

```
add commodity [id] [category] [name] [price] ?detail d1...
```

با این دستور یک کالا با id یکتا با یک اسم و قیمت (عدد حسابی) مشخص در یک کتگوری به فروشگاه اضافه می شود. همچنین هر کالا بعد قسمت این) *detail قسمت می تواند در یک دستور باشد یا نباشد) می تواند تعداد نا مشخصی توضیحات برای کالا داشته باشد که توضیحات در بین دو قرار می گیرند و میتوانند شامل هر کارکتری باشند. همچنین تضمین می شود بین تگ ?detail و گروه اول تنها یک اسپیس قرار دارد و بین گروه ها اسپیی وجود ندارد.

خطاها

- اگر id تکراری باشد باید خطای زیر نمایش داده شود :

```
The id already exists
```

- اگر category در فروشگاه موجود نباشد داریم :

```
The category does not exist
```

افزودن کالا به سبد خرید

add to cart [customer id] [commodity id1,commodity id2,...]

این دستور کالاهایی را که id شان داده شده است را به سبد خرید مشتری با id مشخص اضافه می‌کند. آیدی کالاها با کاما “،” از هم جدا شده‌اند.

خطاها

- اگر id مشتری وجود نداشت :

Customer's id does not exist

اگر آیدی مشتری وجود نداشت لازم به بررسی خطاهای بعدی نیست در غیر اینصورت باید تمام آیدی کالاها بررسی شوند و در صورت وجود خطا، خطای مربوطه نمایش داده شود.

- اگر id کالایی در فروشگاه وجود نداشت :

Id [id] does not exist

که باید در بخش id ، آیدی ناموجود را پرینت کنید.

- اگر یک کالا قبلاً به سبد خرید اضافه شده باشد :

Id [id] has already been added

که باید در بخش id ، آیدی ناموجود را پرینت کنید.

توجه کنید که ترتیب بررسی آیدی کالاها باید مشابه ترتیب آنها در ورودی دستور باشد.

افزایش موجودی کالا

```
increase inventory [commodity id] [amount]
```

این دستور به اندازه عدد amount حسابی) موجودی کالا مشخص شده را افزایش می دهد. اگر id کالا وجود نداشت خطای زیر چاپ شود :

The id does not exist

تغییر اعتبار مشتری

```
change balance [id] [amount]
```

این دستور میزان اعتبار(پول) مشتری با آیدی مشخص را به اندازه amount (یک عدد صحیح) تغییر می دهد. اگر مشتری با id بالا وجود نداشت ، خطای زیر چاپ شود :

The id does not exist

فیلتر کردن کالاهای یک category

```
filter commodity [category] [(lower bound)] to [(upper bound)]
```

این دستور کالا های یک کتگوری را بر اساس قیمت فیلتر می کند. کف و سقف قیمت در بین دو () به عنوان ورودی و شکل هگزادسیمال (مبنا ۱۶) داده می شوند. همچنین توجه شود فرمت ورودی های این حدود به شکل ۰x... است. مثلا دستور پایین کالا هایی با قیمت بین ۱ تا ۱۶ از طبقه category را لیست می کند.

```
filter commodity category (0x1) to (0x10)
```

در خروجی باید نام تمام کالا هایی که قیمتی در بین این کف و سقف دارند را با همان ترتیبی که به فروشگاه اضافه شدند در خطوط جدا و به صورت لیست شده نشان دهید. مثلا :

```
1.banana  
2.ball  
3.laptop
```

خطاها

- اگر کتگوری ورودی در فروشگاه موجود نبود :

The category does not exist

- اگر کف قیمت از سقف قیمت بزرگتر مساوی باشد خطای زیر را باید نشان دهید :

Invalid bounds

توجه شود که حروف موجود در اعداد مبنای ۱۶ به صورت *capital* است. بنابراین اگر حروف به شکل *capital* نباشند ، دستور نامعتبر است. اگر سقف یا کف به فرمتی که گفته شد در دستور نیامدند، دستور نامعتبر است.

خرید کالا

```
buy commodity [customerId] [commodityId] [count] ?comment [[comment]] ?rate
[(rate)]
```

با این دستور مشتری به تعداد ورودی *count* از یک کالا با آیدی مشخص خریداری می کند. همچنین دو ورودی اختیاری *comment* و *rate* وجود دارد که خریدار اگر تمایل داشت از آنها استفاده می کند. *comment* در بین دو [] و *rate* در بین دو () می آید. همچنین برای راحتی کار تضمین می شود که در بین بخش های مختلف قسمت اختیاری دستور بالا (از تگ *comment?* به بعد اسپیس اضافه ای نمی آید).

خطاها

- اگر آیدی مشتری وجود نداشت :

The customer's id does not exist

- اگر آیدی کالا وجود نداشت :

The commodity's id does not exist

- اگر این کالا در سبد خرید مشتری نباشد :

Customer's cart does not include this commodity

- اگر موجودی مشتری به اندازه قیمت کل نبود :

Balance is not enough

- اگر موجودی کالا به اندازه تعداد کالایی که مشتری می‌خواهد نبود :

Insufficient stock

- اگر rate اعداد ۱، ۲، ۳، ۴، ۵ نباشد ، دستور نامعتبر است.

بخش کامنت می‌تواند اسپیس و تب داشته باشد.
توجه شود هر کالا بعد از خرید موفق از سبد خرید حذف می‌شود.

نمایش اطلاعات کالا

commodity info [id]

این دستور اطلاعات یک کالا را مشابه زیر در خطوط جدا جدا نشان می دهد :

```
category :
name :
prize :
count :
detail :
detail :
...
rate :
comment :
comment :
...
```

- منظور از count موجودی این کالا در فروشگاه است .
- هر detail در یک خط می آید و در ابتدای آن : detail می آید. اگر detail برای کالایی تا کنون گذاشته نشده است این بخش نمایش داده نمی شود.
- بخش rate ، میانگین نمراتی است که خریداران به آن کالا می دهند و باید تا دو رقم اعشار در خروجی چاپ شود. اگر کالایی تا کنون نمره ای دریافت نکرده باشد ، این فیلد مقدار ۰ را دارد.
- هر comment در یک خط می آید و در ابتدای آن : comment می آید. اگر کامنتی تا کنون گذاشته نشده است این بخش نمایش داده نمی شود.

- اطلاعات هر بخش با یک اسپیس بعد از : می آید.
- اگر id کالایی در فروشگاه وجود نداشت خطای زیر را خروجی دهید :

The id does not exist

نمایش اطلاعات مشتری

customer info [id]

این دستور اطلاعات یک مشتری را مشابه زیر در خطوط جدا جدا نشان می دهد :

```
name :
balance :
1.commodity :
2.commodity :
...
```

- منظور از balance اعتبار مشتری است.
- اسم هر کالای موجود در سبد خرید در یک خط می آید و در ابتدای آن : -num ber.commodity می آید که number شماره کالا در لیست سبد خرید را مشخص می کند. اگر سبد خرید خالی باشد این بخش نمایش داده نمی شود.
- کالا ها موجود در سبد خرید به ترتیب اضافه شدن به سبد خرید لیست شوند.

- اطلاعات هر بخش با یک اسپیس بعد از : می آید.
- اگر id مشتری در فروشگاه وجود نداشت خطای زیر را خروجی دهید :

```
The id does not exist
```

اتمام برنامه

با این دستور برنامه به اتمام می رسد.

```
end
```

سایر نکات

دستورات نامعتبر

اگر دستوری مشابه فرمت دستورات بالا نباشد و یا گفته شده باشد که حالتی نامعتبر است باید خروجی زیر نمایش داده شود :

```
Invalid command
```

نکات

- در تمامی دستورات id به صورت عددی باید در نظر گرفته شود.
- فیلدی که در یک دستور با علامت ؟ می آید ، اختیاری است یعنی ممکن است در یک دستور باشد و در یک دستور نباشد.

- کارکترهای space white تنها زمانی در یک فیلد (مقادیری که در هر دستور بین دو [] قرار می‌گیرند) می‌آیند که در بخش آن دستور گفته شده باشند.
- قبل و بعد و میان دستورات می‌تواند فضای خالی به تعداد دلخواه وجود داشته باشد.
- زمانی که در اجرای خطایی رخ می‌دهند خطاها باید به ترتیب نوشته شده بررسی شوند و در صورت بروز اولین خطا پیغام مناسب با آن نمایش داده شود و باقی خطاها نباید بررسی شوند (در صورت بروز خطا فقط یک پیام باید چاپ شود). توجه شود که دستور افزودن به سبد خرید از این قاعده مستثنا است و باید خطاهای این دستور به شیوه‌ای که در بخش مربوطه ذکر شده است بررسی شود.

ورودی نمونه ۱

```
add customer 1 john
add category electronicsAndGadgets
add commodity 1 electronicsAndGadgets keyboard 100
start
add customer 1 john
customer info 1
add category electronicsAndGadgets
add commodity 1 electronicsAndGadgets keyboard 200
commodity info 1
add customer 1 james
add category electronicsAndGadgets
add commodity 1 electronicsAndGadgets mouse 100
add commodity 2 electronicsAndGadgets mouse 100 ?detail razermodel : Mamba
Elite
commodity info 2
increase inventory 2 10
commodity info 2
change balance 1 400
customer info 1
add to cart 1 1,2
customer info 1
buy commodity 1 2 2 ?comment [kheili khoob] ?rate (1)
customer info 1
commodity info 2
end
```

خروجی نمونه ۱

```
name : john
balance : 0
category : electronicsAndGadgets
name : keyboard
price : 200
count : 0
```

```
rate : 0.00
The id already exists
The category already exists
The id already exists
category : electronicsAndGadgets
name : mouse
price : 100
count : 0
detail : razer
detail : model : Mamba Elite
rate : 0.00
category : electronicsAndGadgets
name : mouse
price : 100
count : 10
detail : razer
detail : model : Mamba Elite
rate : 0.00
name : john
balance : 400
name : john
balance : 400
1.commodity : keyboard
2.commodity : mouse
name : john
balance : 200
1.commodity : keyboard
category : electronicsAndGadgets
name : mouse
price : 100
count : 8
detail : razer
detail : model : Mamba Elite
rate : 1.00
comment : kheili khoob
```