

بانک

در کتب تاریخی متعددی آمده است که در کشور برزیل (از توابع آبادان کنونی)، سیستم بانکی پیشرفته‌ای وجود داشته و مردم آنجا (من جمله پله‌ی فقید و آقامون رونالدینیو) به داشتن چنین سیستمی افتخار میکردند.



بر اثر حادثه‌ای تاریخی برزیل از آمریکای جنوبی جدا و به زادگاه خود، آبادان برمیگردد.

متأسفانه در حین عبور از تنگه‌ی هرمز، بر اثر پهناوری بیش از حد برزیل، قسمتی از آن به تنگه گیر کرده، شکسته و در آب غرق میشود. نکته‌ی بد اینجاست که سیستم بانکی برزیل در آن قسمت غرق شده قرار داشت و اکنون برزیل باقیمانده سیستم بانکی ندارد.

در این سوال ما از شما می‌خواهیم تا سیستم بانکی برزیل را مجدداً پیاده سازی کنید. خوشبختانه نقشه‌ی این سیستم بانکی از بین نرفته و در این سوال، شما باید دقیقاً طبق همین نقشه (که در اصل به آن UML میگویند) سیستم بانکی برزیل را مجدداً پیاده سازی کنید.

این سیستم بانکی دارای اجزای زیر است:

بانک:

در برزیل هر بانک یک نام مخصوص به خود دارد و خدمات افتتاح حساب و اعطای وام را ارائه می‌کند.

مشتری:

هر مشتری یک اسم مخصوص به خود دارد و هیچ دو مشتری‌ای اسم یکسان ندارند. نمره‌ی منفی ویژگی دیگریست که مشتری‌ها دارند. در مورد نمره‌ی منفی در قسمت وام خواهید خواند.

هر مشتری در ابتدا مقداری پول در گاوصندوق خانه‌اش دارد. او میتواند یک مقدار از این پول را در هر بانکی در قالب یک حساب، سپرده‌گذاری کند. با تمام شدن مدت تعیین شده برای حساب، مبلغ سپرده گذاری شده همراه با سود آن به صورت اتوماتیک (!) به گاوصندوق مشتری برمی‌گردد.

اگر مشتری زودتر از مدت تعیین شده برای بازپس گرفتن سپرده‌اش به بانک مراجعه کند، تنها مبلغ اصلی (بدون سود) را دریافت میکند و به گاوصندوقش برمی‌گرداند.

با پس گرفته شدن پول یک حساب یا به پایان رسیدن مدت تعیین شده‌ی حساب، آن حساب از بین میرود.

حساب:

سه نوع حساب در بانک‌های برزیل وجود دارد؛ حساب‌های کوتاه‌مدت، بلندمدت و ویژه. هر حساب شماره، میزان سپرده‌ی اولیه، درصد سود و مدت تعیین شده دارد. پس از سپری شدن زمان تعیین شده، درصد سود به علاوه‌ی یک میشود و در میزان سپرده‌ی اولیه ضرب میشود، آنگاه به گاوصندوق صاحب حساب بازمی‌گردد.

درصد سود حساب‌های کوتاه مدت، بلندمدت و ویژه به ترتیب 10٪، 30٪ و 50٪ است. نوع حساب (کوتاه مدت، بلند مدت و ویژه) و مدت حساب در هنگام ساختن حساب توسط مشتری تعیین میشود.

نکته‌ی دیگر اینکه شماره‌ی هر حسابی که هر مشتری ایجاد میکند، یکی بیشتر از تعداد حساب‌هایی می‌شود که همان مشتری از بدو خلقت تا الان ساخته است؛ برای مثال اگر یک مشتری پنجمین حساب خود را ایجاد کند، شماره‌ی آن حساب 5 میشود (یعنی ممکن است در بانکی دو حساب با شماره‌ی یکسان وجود داشته باشد ولی مطمئناً آن دو حساب متعلق به دو نفر مختلف هستند).

وام:

هر وام درصد سود و مقدار دارد. دو نوع وام داریم: وام 6 قسطه و 12 قسطه. بعد از ثبت درخواست، وام به گاو صندوق مشتری وارد میشود.

مکانیزم پرداخت اقساط به این صورت است که در هر واحد زمانی یک قسط به مقدار زیر از گاو صندوق مشتری کم میشود:

$$\text{each payment value} = \frac{\text{loan initial value} * (\% \text{interest} + 1)}{\text{payments counts}}$$

اگر مشتری در گاو صندوق خود به مقدار قسط پول نداشته باشد، هیچ مقداری از گاو صندوق او کم نشده، تعداد اقساط باقیمانده اش ثابت می ماند و یک نمره منفی می گیرد.

بدی نمره منفی این است که اگر مشتری ای حداقل پنج نمره منفی داشته باشد دیگر وامی به او تعلق نمی گیرد.

به این نکته هم توجه داشته باشید که ترتیب کم شدن اقساط وامها براساس ترتیب به وجود آمدن وامهاست؛ یعنی مثلاً اگر وام x زودتر از وام y در ورودی آمده باشد، قسط وام x زودتر از قسط وام y از حساب مشتریان کم میشود.

عملیات هایی که در سیستم بانکی برزیل پشتیبانی میشوند به شرح زیر هستند:

افزودن یک مشتری با نام A و N واحد پول اولیه (که A یک اسم یک یا چند بخشی است و N یک عدد طبیعیست):

Add a customer with name A and N unit initial money.

افزودن بانک با نام M (که M یک اسم یک یا چند بخشی است) مثلاً "golabi" یا "kaka khan<3" دو اسم مجاز هستند):

Create bank M.

افزودن یک حساب کوتاه مدت، بلند مدت یا ویژه در بانک M با مدت زمان T و میزان سپرده‌ی اولیه‌ی D توسط مشتری A:

Create a [KOOTAHIBOLANIVIZHE] account for A in M, with duration T and in

- اگر بانک با نام M وجود نداشت، پیغام زیر را چاپ کنید:

In dige banke koodoom keshvarie?

- اگر مشتری A کمتر از D واحد درون گاوصندوق خود پول داشت، پیغام زیر را چاپ کنید:

Boro baba pool nadari!

- (خطاها را به همین ترتیب چک کنید و خروجی دهید)

برداشتن تمام پول شخص A از حساب شماره‌ی N:

Give A's money out of his account number N.

- اگر مشتری حساب با همچنین شماره‌ای نداشت، پیغام زیر را چاپ کنید:

Chizi zadi?!

پرداخت یک وام با مقدار X واحد و سود Y% و اقساط 6 یا 12 واحدی از بانک M به شخص A:

Pay a X unit loan with Y% interest and [6|12] payments from M to A.

- چنانچه بانک با نام M وجود نداشت، پیغام زیر را چاپ کنید:

Gerefti maro nesfe shabi?

- چنانچه A حداقل 5 نمره‌ی منفی داشته باشد، پیغام زیر را چاپ کنید:

To yeki kheyli vazet bade!

- (خطاها را به همین ترتیب چک کنید و خروجی دهید).

از ابتدای خلقت X واحد میگذرد:

Pass time by X unit.

میزان پول موجود در گاوصندوق مشتری A را چاپ کنید:

Print A's GAVSANDOOGH money.

تعداد نمره های منفی A را چاپ کنید:

Print A's NOMRE MANFI count.

آیا شخص A در بانک M حساب فعال دارد یا نه؟

Does A have active account in M?

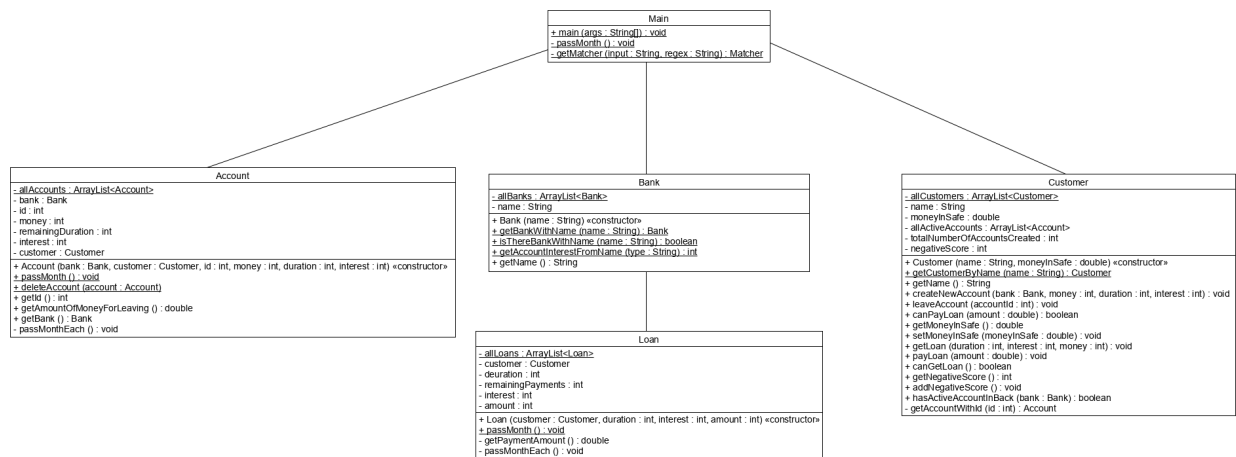
- در صورتی که A در بانک M حساب فعالی داشته باشد عبارت "yes" و در غیر این صورت عبارت "no" را چاپ کنید.

انجام عملیات‌های بانکی خاتمه می‌یابد:

Base dige, berid khonehatoon.

چند نکته:

- پرداخت اقساط وام و کم شدن از مدت تعیین شده برای حسابها، در پایان هر واحد زمانی اتفاق می افتد و سایر اعمال از جمله "واریز سپرده ی حسابها پس از تمام شدن مدت حساب"، در ابتدای هر واحد زمانی رخ میدهد.
- به نقطه ها و علائم نگارشی آخر دستورات و جملات توجه کنید. در دستورات ورودی و خروجی عیناً همانها آورده می شود.
- تضمین می شود تنها خطاهایی که در بالا برای آنها پاسخی در نظر گرفته شده رخ می دهند و خطاهای دیگری در دستورات ورودی رخ نمیدهد. ترتیب خطاها را همانگونه که آورده شده چک کنید و خروجی دهید.
- در هنگام چاپ اعداد در خروجی، آنها را قطع و چاپ کنید؛ مثلاً اگر میزان پول موجود در گاو صندوق مشتری یکس 12.34 واحد باشد، شما باید عدد 12 را چاپ کنید.
- کدی که میزنید حتماً باید براساس UML ای که در زیر مشاهده می کنید پیاده سازی شود. این موضوع در تحویل حضوری بررسی خواهد شد.



ورودی نمونه

Add a customer with name agha tizi and 100 unit initial money.
 Add a customer with name akbar and 50 unit initial money.
 Add a customer with name mika the real kaka and 3 unit initial money.
 Create bank melli.
 Create bank mehr.
 Create a VIZHE account for agha tizi in mellat, with duration 3 and ini
 Create a VIZHE account for agha tizi in melli, with duration 3 and init

Pay a 50 unit loan with %17 interest and 12 payments from mellat to mika
Pay a 50 unit loan with %17 interest and 12 payments from melli to mika
Create a BOLAN account for mika the real kaka in melli, with duration 3
Pass time by 20 unit.
Print agha tizi's GAVSANDOOGH money.
Print akbar's GAVSANDOOGH money.
Print mika the real kaka's GAVSANDOOGH money.
Print mika the real kaka's NOMRE MANFI count.
Base dige, berid khonehatoon.

خروجی نمونه

In dige banke koodoom keshvarie?
Gerefti maro nesfe shabi?
135
50
9
3

سپاتیفای

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

مرد هلندی که به تازگی با زبان جاوا آشنا شده است، تصمیم گرفته است نسخه ساده‌شده‌ای از نرم‌افزار موردعلاقه‌اش یعنی spotify را با کمک این زبان پیاده‌سازی کند. از آنجایی که او هنوز درس AP را برنداشته است از شما کمک خواسته است تا با استفاده از UML class diagram داده شده این برنامه را برای او بنویسید. فایل UML را می‌توانید از [اینجا](#) مشاهده کنید.



پیاده‌سازی شما باید کاملاً مطابق با UML داده شده باشد. در غیر این صورت نمره‌ای به کد شما تعلق نمی‌گیرد.

منوی ثبت‌نام و لاگین

هنگام شروع اجرای برنامه در این منو قرار داریم. دستورهای این منو شامل موارد زیر می‌باشند.

ثبت نام به عنوان کاربر

نام کاربری می‌تواند تنها شامل حروف کوچک و بزرگ انگلیسی و اعداد باشد، همچنین رمز عبور از حروف بزرگ و کوچک انگلیسی و اعداد تشکیل شده و باید حداقل یک حرف کوچک، یک حرف بزرگ و یک عدد داشته باشد در غیر این صورت به اندازه کافی امن نیست.

```
register as user -u <username> -p <password>
```

در صورت اجرای موفقیت آمیز کاربر با این مشخصات به کاربرهای برنامه اضافه می‌شود و پیام زیر نمایش داده شود.

```
user registered successfully
```

خطاها:

در صورتی که کاربری با نام کاربری انتخاب شده از قبل وجود داشته باشد پیام زیر نمایش داده شود.

```
username already exists
```

در صورتی که رمز عبور انتخاب شده به اندازه کافی امن نباشد پیام زیر نمایش داده شود.

```
password is not strong enough
```

ثبت نام به عنوان هنرمند

نام کاربری و رمز عبور انتخاب شده لازم است شرایطی مانند ثبت نام کاربر داشته باشند. nickname می‌تواند از حروف کوچک و بزرگ انگلیسی، اعداد و کاراکتر فاصله تشکیل شده باشد.

```
register as artist -u <username> -p <password> -n <nickname>
```

در صورت اجرای موفقیت‌آمیز هنرمندی با این مشخصات به برنامه افزوده می‌شود و پیام زیر نمایش داده شود.

```
artist registered successfully
```

خطاها شامل قوی نبودن رمزعبور و تکراری بودن نام کاربری بین هنرمندان موجود می‌باشد. پیام مرتبط با این خطاها مشابه با ثبت‌نام کاربر می‌باشد.

ورود به عنوان کاربر

```
login as user -u <username> -p <password>
```

در صورت اجرای موفقیت‌آمیز برنامه وارد منوی کاربر می‌شود و پیام زیر نمایش داده می‌شود.

```
user logged in successfully
```

خطاها:

در صورتی که کاربری با نام کاربری گفته شده وجود نداشته باشد پیام زیر نمایش داده می‌شود.

```
username doesn't exist
```

در صورتی که رمزعبور واردشده نادرست است پیام زیر نمایش داده شود.

```
password is wrong
```

ورود به عنوان هنرمند

```
login as artist -u <username> -p <password>
```

در صورت اجرای موفقیت‌آمیز برنامه وارد منوی هنرمند می‌شود و پیام زیر نمایش داده می‌شود. خطاهای ممکن برای این دستور و پیام متناظر با آنها مشابه ورود به عنوان کاربر هستند.

```
artist logged in successfully
```

منوی کاربر

دستورهای این منو به شرح زیر می‌باشند.

نمایش پلی لیست‌ها

```
show playlists
```

با وارد کردن این دستور نام پلی لیست های کاربر log in شده، به ترتیب الفبا و هرکدام در یک خط، چاپ می‌شود.

نمایش قطعات موردعلاقه

```
show liked tracks
```

با وارد کردن این دستور نام قطعات مورد علاقه کاربر، به ترتیب الفبا و هرکدام در یک خط، چاپ می‌شوند.

نمایش صف

منظور از صف لیست قطعاتی است که به ترتیب برای کاربر پخش می‌شوند. اضافه و حذف شدن قطعات به این لیست توسط خود کاربر انجام می‌شود.

```
show queue
```

با وارد کردن این دستور نام قطعات صف به ترتیب وارد شدن به صف و هرکدام در یک خط چاپ می‌شوند.

افزودن قطعه به صف

با اجرای این دستور قطعه به انتهای صف کاربر افزوده می‌شود.

```
add -t <trackname> to queue
```

نام قطعه از حروف کوچک و بزرگ انگلیسی، اعداد و کاراکتر فاصله تشکیل شده است. در صورت اجرای موفقیت آمیز پیام زیر نمایش داده می‌شود.

```
track added to queue successfully
```

خطاها:

در صورتی که قطعه ای با نام گفته شده وجود نداشته باشد پیام زیر نمایش داده می‌شود.

```
no such track
```

افزودن قطعه به قطعات مورد علاقه

```
like track -t <trackname>
```

در صورت اجرای موفقیت آمیز پیام زیر نمایش داده می‌شود.

```
liked track successfully
```

خطای وجود نداشتن قطعه ای با این نام و پیام متناظر با آن مشابه با افزودن قطعه به صف می‌باشد. در صورتی که قطعه از قبل عضو قطعات مورد علاقه بوده است پیام زیر نمایش داده می‌شود.

```
track is already liked
```

حذف قطعه از صف

```
remove -t <trackname> from queue
```

در صورت اجرای موفقیت آمیز قطعه با نام گفته شده از صف کاربر حذف می‌شود و پیام زیر نمایش داده می‌شود.

```
track removed from queue successfully
```

خطاها:

در صورتی که قطعه ای با این نام در صف موجود نباشد پیام زیر نمایش داده می‌شود.

```
no such track in queue
```

ممکن است قطعه موردنظر بیش از یک بار در صف باشد، لازم است تمام تکرارهای آن از صف حذف شوند

حذف قطعه از قطعات موردعلاقه

```
remove -t <trackname> from liked tracks
```

در صورت اجرای موفقیت آمیز قطعه با نام گفته شده از لیست قطعات موردعلاقه حذف می‌شود و پیام زیر نمایش داده می‌شود.

```
track removed from liked tracks successfully
```

خطاها:

در صورتی که قطعه ای با این نام در بین قطعات موردعلاقه کاربر نباشد پیام زیر نمایش داده می‌شود.

```
no such track in liked tracks
```

معکوس کردن صف

با این دستور کاربر میتواند ترتیب قطعات بخشی از صف را برعکس کند. در این دستور شروع و پایان بازه (start و end) عدد هستند .

reverse order of queue from <start> to <end>

در صورت اجرا شدن پیام زیر را نمایش دهید.

order of queue reversed successfully

خطاها: در صورتی که صف خالی باشد، پیام زیر نمایش داده شود.

queue is empty

اگر اعداد شروع و پایان در بازه ۱ تا تعداد قطعات صف نباشند، باید پیام زیر نمایش داده شود. همچنین باید بررسی شود که $start < end$ و در غیر اینصورت بازه نامعتبر است.

invalid bounds

ساخت پلی لیست

create -p <playlistName>

نام پلی لیست میتواند شامل حروف کوچک و بزرگ انگلیسی، اعداد و کاراکتر فاصله باشد. در صورت اجرای موفقیت آمیز پلی لیست با نام گفته شده برای کاربر ساخته میشود و پیام زیر نمایش داده میشود. *برای سادگی تمام پلی لیستهای ساخته شده را public در نظر می گیریم *

playlist created successfully

خطاها:

در صورتی که پلی لیستی با این نام وجود داشته باشد پیام زیر نمایش داده می‌شود.

```
playlist name already exists
```

حذف پلی لیست

```
delete -p <playlistName>
```

در صورت اجرای موفقیت‌آمیز پلی لیست با نام گفته شده از مجموعه پلی لیست ها حذف می‌شود و پیام زیر نمایش داده می‌شود.

```
playlist deleted successfully
```

خطاها:

در صورتی که پلی لیستی با این نام وجود نداشته باشد یا سازنده آن کاربر log in شده نباشد پیام زیر نمایش داده می‌شود.

```
user doesn't own such playlist
```

دنبال کردن کاربر دیگر

برای سادگی، برای هر کاربر تنها تعداد دنبال‌کننده ها و دنبال شونده‌هایش نگهداری می‌شوند و نیازی به نگهداری اسامی دنبال کننده ها و دنبال‌شونده هایش نیست. تضمین می‌شود هیچ کاربری بیش از یک بار اقدام به دنبال کردن کاربر دیگر نمی‌کند.

```
follow user -u <username>
```

در صورت اجرای موفقیت‌آمیز کاربر فعلی کاربری با نام کاربری وارد شده را دنبال می‌کند و پیام زیر نمایش داده می‌شود.

added user to followings

خطاها:

در صورتی که کاربری با نام کاربری وارد شده وجود نداشته باشد پیام زیر نمایش داده می‌شود.

no such user

در صورتی که کاربر نام کاربری خود را وارد کند، پیام زیر نمایش داده می‌شود.

you can't follow yourself

دنبال کردن هنرمند

برای سادگی، برای هر هنرمند تنها تعداد دنبال کننده هایش نگهداری می‌شود و نیازی به نگهداری اسامی آنها نیست.

تعداد following برای هر کاربر برابر با جمع تعداد کاربرها و هنرمندهایی است که دنبال می‌کند

follow artist -u <username>

در صورت اجرای موفقیت‌آمیز پیام زیر نمایش داده می‌شود.

added artist to followings

خطاها:

در صورتی که هنرمندی با این نام کاربری وجود نداشته باشد پیام زیر نمایش داده می‌شود.

no such artist

آنفالو کردن کاربر دیگر

تضمین می‌شود این دستور برای کاربری که جز following ها نیست فراخوانی نمی‌شود.

```
unfollow user -u <username>
```

در صورت اجرای موفقیت‌آمیز پیام زیر نمایش داده می‌شود.

```
user unfollowed successfully
```

خطاها:

در صورتی که کاربری با نام کاربری گفته شده وجود نداشته باشد پیام زیر نمایش داده می‌شود.

```
no such user
```

در صورتی که نام کاربری وارد شده با نام کاربری کاربر login شده یکسان بود پیام زیر نمایش داده می‌شود.

```
you can't unfollow yourself
```

آنفالو کردن هنرمند

تضمین می‌شود این دستور برای هنرمندی که جز following ها نیست فراخوانی نمی‌شود.

```
unfollow artist -u <username>
```

در صورت اجرای موفقیت‌آمیز پیام زیر نمایش داده می‌شود.

```
artist unfollowed successfully
```

خطاها:

در صورتی که هنرمندی با نام کاربری گفته شده وجود نداشته باشد پیام زیر چاپ می‌شود.

no such artist

رفتن به منوی پلی لیست

go to playlist menu -p <playlistName>

در صورت اجرای موفقیت‌آمیز این دستور برنامه وارد منوی پلی لیست با نام گفته شده می‌شود و پیام زیر نمایش داده می‌شود.

entered playlist menu successfully

خطاها:

در صورتی که پلی لیستی با نام گفته شده وجود نداشته باشد پیام زیر نمایش داده می‌شود.

no such playlist

نمایش اطلاعات یک قطعه

show track info -t <trackName>

با اجرای این دستور، نام قطعه، تایپ قطعه (آهنگ یا پادکست)، مدت زمان آن برحسب ثانیه، تاریخ انتشار آن با فرمت yyyy/mm/dd و nickname هنرمند منتشرکننده آن، همه در یک خط و با یک فاصله بین شان چاپ می‌شود.

در صورتی که قطعه ای با این نام وجود نداشته باشد پیام زیر نمایش داده می‌شود.

no such track

نمایش اطلاعات یک پلی لیست

```
show playlist info -p <playlistName>
```

با اجرای این دستور نام پلی لیست، نام کاربر سازنده آن و مدت زمان پلی لیست همگی در یک خط و با یک فاصله بین شان چاپ می‌شوند.

در صورتی که پلی لیستی با این نام وجود نداشته باشد پیام زیر نمایش داده می‌شود.

```
no such playlist
```

نمایش اطلاعات یک هنرمند

```
show artist info -u <username>
```

با اجرای این دستور نام کاربری هنرمند، nickname او، تعداد دنبال کننده و رنگ او، همگی در یک خط و با یک فاصله بین شان چاپ می‌شوند.

در صورتی که هنرمندی با این نام کاربری وجود نداشته باشد پیام زیر نمایش داده می‌شود.

```
no such artist
```

نمایش اطلاعات یک کاربر

```
show user info -u <username>
```

با اجرای این دستور نام کاربری کاربر، تعداد follower ها، تعداد following ها و تعداد پلی لیست های او به ترتیب در یک خط و با یک فاصله بین شان چاپ می‌شوند.

در صورتی که کاربری با این نام کاربری وجود نداشته باشد پیام زیر نمایش داده می‌شود.

no such user

منوی هنرمند

در صورتی که به عنوان یک هنرمند لاگین کرده باشید وارد این منو می‌شوید.

نمایش قطعات

show tracks

با اجرای این دستور نام قطعات منتشرشده توسط هنرمند به ترتیب جدیدترین تاریخ انتشار و سپس به ترتیب الفبا، هر کدام در یک خط نمایش داده می‌شوند.

نمایش آهنگ‌ها

قطعات به دو دسته آهنگ و پادکست تقسیم می‌شوند که فیلد type برای آنها نشان‌دهنده این است که عضو کدام دسته هستند.

show songs

با اجرای این دستور نام آهنگ‌های منتشرشده توسط هنرمند، به ترتیب جدیدترین تاریخ انتشار و سپس به ترتیب الفبا، هر کدام در یک خط نمایش داده می‌شوند.

نمایش پادکست‌ها

show podcasts

با اجرای این دستور نام پادکست‌های منتشرشده توسط هنرمند به ترتیب جدیدترین تاریخ انتشار و سپس به ترتیب الفبا، هر کدام در یک خط نمایش داده می‌شوند.

افزودن قطعه

```
release -n <trackName> -t <type> -d <duration> -r <releaseDate>
```

منظور از type آهنگ یا پادکست بودن قطعه است. به جای <type> تنها یکی از دو عبارت song یا podcast باید قرار بگیرند. منظور از duration زمان قطعه بر حسب ثانیه است. تاریخ انتشار تنها به فرمت yyyy/mm/dd قابل قبول است. (مثال: 2022/12/02) با اجرای موفقیت‌آمیز این دستور قطعه ای با مشخصات گفته شده به قطعات منتشرشده توسط هنرمند افزوده می‌شود و پیام زیر نمایش داده می‌شود.

```
track released successfully
```

خطاها:

در صورتی که قطعه ای با نام انتخاب شده پیش از این منتشر شده بود (توسط این هنرمند یا هنرمندهای دیگر) پیام زیر نمایش داده می‌شود.

```
track name already exists
```

تعداد دنبال‌کننده‌ها

```
num of followers
```

با وارد کردن این دستور تعداد دنبال کننده های هنرمند چاپ می‌شود.

رتبه

```
get rank
```

با وارد کردن این دستور رتبه هنرمند بین هنرمندان بر اساس تعداد دنبال کننده چاپ می‌شود.

منوی پلی لیست

نمایش قطعه‌ها

```
show tracks
```

با اجرای این دستور نام قطعات موجود در پلی لیست به ترتیب جدیدترین تاریخ انتشار و سپس به ترتیب الفبا، هرکدام در یک خط چاپ می‌شوند.

نمایش مدت زمان

```
show duration
```

با اجرای این دستور مدت زمان پلی لیست که برابر با مجموع مدت زمان قطعات تشکیل‌دهنده آن است، برحسب ثانیه چاپ می‌شود.

اضافه کردن قطعه

```
add -t <trackName>
```

در صورت اجرای موفقیت‌آمیز قطعه به پلی لیست اضافه می‌شود و پیام زیر نمایش داده می‌شود.

```
track added to playlist successfully
```

خطاها: در صورتی که کاربر login شده صاحب این پلی لیست نباشد پیام زیر نمایش داده می‌شود.

```
user doesn't own this playlist
```

در صورتی که قطعه ای با نام گفته شده وجود نداشته باشد پیام زیر نمایش داده می‌شود.

no such track

در صورتی که این قطعه از قبل در پلی لیست بوده است پیام زیر نمایش داده می‌شود.

track is already in the playlist

حذف قطعه از پلی لیست

```
remove -t <trackName>
```

در صورت اجرای موفقیت‌آمیز قطعه با نام گفته شده از پلی لیست حذف می‌شود و پیام زیر نمایش داده می‌شود.

track removed from playlist successfully

خطاها:

در صورتی که قطعه ای با نام گفته شده در پلی لیست موجود باشد پیام زیر نمایش داده می‌شود.

no such track in playlist

تکات: *در تمامی منوها با وارد کردن دستور back به منوی قبلی می‌رویم. در صورتی که در منوی ثبت نام back وارد شود برنامه تمام می‌شود* *در تمامی منوها در صورتی که دستور وارد شده مطابق با دستورات آن منو نباشد پیام زیر چاپ می‌شود.*

invalid command

در تمامی منوها با زدن دستور show menu name باید نام منوی فعلی چاپ شود.

ورودی نمونه


```
register as artist -u drake123 -p drake135 -n DRAKE
register as artist -u drake123 -p Drake135 -n DRAKE
login as artist -u drake123 -p Drake135
release -n t1 -t song -d 100 -r 2020/12/12
release -n t2 -t song -d 130 -r 2022/11/11
back
register as user -u rihanna -p Rhn789
login as user -u rihanna -p Rhn789
follow artist -u drake123
add -t t1 to queue
add -t t2 to queue
like track -t t1
show liked tracks
show queue
back
exit
```

خروجی نمونه

```
password is not strong enough
artist registered successfully
artist logged in successfully
track released successfully
track released successfully
user registered successfully
user logged in successfully
added artist to followings
track added to queue successfully
track added to queue successfully
liked track successfully
t1
t1
t2
```

پوکمان

آقای عماد از معدود نفرات در ایران است که به بازی پوکمان علاقه داشته و سالهاست در تلاش است که بازی پوکمان را به عنوان پروژه به دانشجویان بدهد و همیشه شکست خورد. تا این که نوبت به تمرین عید ای پی این ترم رسید و بالاخره نظر خود را بر کرسی نشاند. حال این بازی را برای او بسازید تا بتواند پوکمان بازی کند و شاید بالاخره متوجه شود که بازی جالبی نیست.



هدف از این سوال، پیاده‌سازی بازی کارتی مشابه با بازی پوکمان است که بخش‌هایی از آن به خاطر سهولت در پیاده‌سازی، در نظر گرفته نشده‌اند.

در هر یک از بخش‌های برنامه اگر عبارتی وارد شد، به ترتیب ذکر شده خطاها را چک کنید و فقط اولین خطای رخ داده را چاپ کنید. اگر که دستور متناظر با آن وجود نداشت عبارت

invalid command

را خروجی دهید.

همچنین در هر منویی که باشیم با دستور

show current menu

می‌توانیم منویی که درحال حاضر در آن هستیم را مشاهده کنیم که یکی از پنج حالت زیر را دارد:

login menu
profile menu
shop menu
main menu
game menu

این بازی شامل پنج منوی مختلف زیر است:

۱. لاگین منو
۲. پروفایل منو
۳. منوی فروشگاه
۴. منوی بازی
۵. منوی اصلی

که در ادامه هریک از آنها را توضیح خواهیم داد.

- در هر یک از منوهای فروشگاه یا پروفایل، می‌توانیم با دستور

back

به منوی اصلی بازگردیم.

- همچنین در منوی اصلی با دستور

logout

می‌توانیم به منوی لاگین بازگردیم و از نقش خود خارج شویم.

- در منوی لاگین با استفاده از دستور

exit

به طور کامل از بازی خارج می‌شویم.

کلیات بازی:



پوکمان یک بازی turn-based و تک به تک است؛ به این صورت که هر بازیکن قبل از شروع بازی ۱۲ کارت از میان تمام کارتهایی که دارد را تحت عنوان deck برای خود انتخاب می‌کند و وارد بازی می‌شود. هدف هر بازیکن این است که با ابزارهای مختلف مثل کارتهای انرژی و تکنیکهای دیگر، تمام پوکمانهای حریف را نابود کند تا برنده شود.

در این بازی کارتی ما ۳ کلاس اصلی کارت پوکمان به نامهای Fire و Water و Plant داریم که هر یک از آنها شامل دو نوع مختلف پوکمان می‌شود. همچنین ۲ نوع کارت انرژی نیز داریم. کارتهای انرژی به پوکمانی که تخصیص داده شده‌اند کمک می‌کنند.

- کارتهای پوکمان، کارتهای اصلی بازی هستند که در هنگام بازی کردن در یک دست برای انجام یک action از آنها استفاده می‌کنیم.

- در هنگام شروع کردن بازی هر فرد باید ۱۲ کارت را از میان کارتهایی که دارد انتخاب کرده باشد و در بازی بیاورد. دقت کنید که در کل ۶ نوع پوکمان مختلف (۲ نوع از هر کلاس) داریم و از هر کدام از آنها حداکثر یکی را می توان به داخل بازی آورد؛ بنابراین هر فرد در دست خود حداکثر ۶ پوکمان خواهد داشت.

لاگین منو

ثبت نام

در این منو هر کاربر می تواند با دستور

```
register username <username> password <password> email <email>
```

به عنوان کاربر جدید به بازی اضافه شود.

به نکات زیر در رابطه با این دستور توجه کنید:

۱. بین هر کدام از عبارت‌های موجود در آن می‌تواند تعداد دلخواهی فاصله یا کاراکتر اسپیس بیاید.
۲. بخش username تنها می‌تواند شامل حروف کوچک و بزرگ انگلیسی و همچنین آندرلاین باشد.
۳. بخش password باید دارای حداقل ۶ کاراکتر و حداکثر ۱۸ کاراکتر باشد و همچنین حداقل یک کاراکتر از میان کاراکترهای @!&^\$# را داشته باشد. رمز عبور نباید با کاراکتری غیر از حروف کوچک یا بزرگ انگلیسی شروع شود و نباید دارای فاصله یا کاراکتر اسپیس باشد.
۴. بخش ایمیل باید به صورت `address@something.com` باشد و بخش آدرس آن نیز تنها می‌تواند دارای حروف کوچک و بزرگ انگلیسی، اعداد و "." باشد. ایمیل نباید بیش از یکبار کاراکتر "." را داشته باشد و همچنین بخش something تنها می‌تواند دارای حروف کوچک انگلیسی باشد.

خطاها:

اگر username با فرمت گفته شده نبود، عبارت

username format is invalid

را چاپ کنید.

اگر username داده شده تکراری بود، عبارت زیر را چاپ کنید.

username already exists

اگر password داده شده کمتر از ۶ کاراکتر یا بیشتر از ۱۸ کاراکتر داشت، عبارت زیر را چاپ کنید.

password length too small or short

اگر password داده شده کاراکترهایی از بین @!&^\$ را نداشته باشد، باید عبارت زیر را چاپ کنید.

your password must have at least one special character

اگر رمز عبور با کاراکتری غیر از حروف کوچک یا بزرگ انگلیسی شروع شود، باید عبارت زیر را چاپ کنید.

your password must start with english letters

اگر email با فرمت گفته شده پایان نمی‌یافت، باید عبارت زیر را چاپ کنید.

email format is invalid

اگر بخش آدرس دارای کاراکتری به جز کاراکترهای گفته شده باشد و یا بیش از یکبار از کاراکتر . استفاده شده‌باشد، باید عبارت زیر را چاپ کنید.

you can't use special characters

اگر هیچ یک از خطاها وجود نداشت، کاربر با موفقیت ثبت نام می‌کند. مقادیر اولیه‌ی coin برابر با ۳۰۰ و experience برابر با ۰ را برای او لحاظ کنید و پیام زیر را چاپ کنید، سپس دوباره به Login Menu بر

می‌گردیم.

user registered successfully

ورود:

کاربر می‌تواند با دستور

```
login username <username> password <password>
```

وارد شود و بین بخش‌های مختلف آن می‌تواند تعداد دلخواهی کاراکتر اسپیس بیاورد.

خطاها:

اگر کاربری با این username وجود نداشته باشد، باید خطای زیر را نمایش دهید.

username doesn't exist

اگر password اشتباه باشد، باید عبارت زیر را چاپ کنید.

password is incorrect

در صورتی که کاربر با خطایی مواجه نشد، با موفقیت وارد می‌شود و پس از چاپ پیام زیر، وارد منوی اصلی می‌شود.

user logged in successfully

منوی اصلی:

در این منو می‌توانیم وارد منوی پروفایل و یا منوی فروشگاه شویم. شروع کردن بازی جدید نیز از طریق این منو انجام می‌شود.

- رفتن به منوی فروشگاه:

با استفاده از دستور زیر می‌توانیم وارد منوی فروشگاه شویم.

go to shop menu

- رفتن به منوی پروفایل:

با استفاده از دستور زیر می‌توانیم وارد منوی پروفایل شویم.

go to profile menu

- شروع کردن بازی جدید:

با استفاده از دستور زیر می‌توانیم یک بازی جدید را با کاربر گفته شده ایجاد کنیم.

start new game with <username>

خطاها:

۱. اگر چنین کاربری وجود نداشت خطای زیر باید نمایش داده شود.

username is incorrect

۲. اگر از هر یک از افراد ۱۲ کارت در دست خود برای شروع بازی نداشت پیام زیر را نمایش دهید. توجه کنید که این پیام حداکثر یک بار چاپ می‌شود و ابتدا باید deck کاربری که در حال حاضر login کرده‌است، بررسی شود.

<username> has no 12 cards in deck

منوی پروفایل:

این منو مربوط به ویژگی‌های کاربر از جمله تعداد سکه‌ها، تجربه و... است. همچنین کاربر می‌تواند دستی که برای بازی پیش‌رو می‌خواهد داشته باشد را از میان کارتهای خود در این بخش انتخاب کند.

- نشان دادن تعداد coinها با استفاده از دستور زیر انجام می‌شود.

show coins

پس از این دستور، کاربر می‌تواند به فرم زیر تعداد coinهایی که دارد را مشاهده کند.

number of coins:<coins>

- نشان دادن experience با استفاده از دستور زیر انجام می‌شود.

show experience

پس از این دستور کاربر می‌تواند به فرم زیر، experience خود را مشاهده کند.

experience:<experience>

- نشان دادن کارتهای موجود در سبد خود با استفاده از دستور زیر انجام می‌شود.

show storage

پس از این دستور کاربر می‌تواند تمام کارتهایی را که از ابتدا خریده است و کماکان دارد را ببیند.

لازم به ذکر است کارتهایی که به دست اضافه شده‌اند نیز نشان داده می‌شوند. دقت کنید کارتها با ترتیبی که خریده شده‌اند، در قالب زیر نمایش داده می‌شوند.

```
1.<card type> <card name> <value>
2.<card type> <card name> <value>
....
```

بخش **card type** می‌تواند یکی از بین **fire، water ، plant** و یا **energy** باشد. **card name** نیز اسم آن نوع از کارت است. بخش **value** نیز نشان دهنده‌ی تعداد **coin** مورد نیاز برای خرید آن است. تمام کاراکترهای خروجی **lowercase** هستند.

- کاربر برای اضافه کردن یک کارت به دست خود از میان کل کارتهایی که دارد، می‌تواند دستور زیر را وارد کند.

`equip card <card name> to my deck`

توجه کنید همان‌طور که در پایین‌تر نیز ذکر شده است، برای کارتهای **Energy** در بخش اسم کارت، عبارت **pink** یا **yellow** می‌آید.

خطاها:

۱. اگر اسم کارت اشتباه بود باید خطای زیر را نمایش دهید.

`card name is invalid`

۲. اگر چنین کارتی در میان کارتهایی که عضو **deck** نشده‌اند وجود نداشت، باید عبارت زیر را چاپ کنید:

`you don't have this type of card`

۳. اگر تاکنون ۱۲ کارت به **deck** اضافه شده بود (**deck** پر شده‌بود)، خطای زیر نمایش داده می‌شود.

`your deck is already full`

۴. اگر تا کنون از آن نوع پوکمان خاص به **deck** اضافه کرده بودیم، باید خطای زیر نمایش داده شود. (توجه کنید که در هر دست حداکثر یک کارت از هر نوع پوکمان داریم ولی در هنگام خرید، چنین محدودیتی وجود ندارد.)

`you have already added this type of pokemon to your deck`

در صورتی که با خطایی روبه‌رو نشدیم، کارت با موفقیت به دست فرد اضافه می‌شود و پیام زیر چاپ می‌گردد.

card <card name> equipped to your deck successfully

- کاربر با استفاده از دستور زیر

unequip card <card name> from my deck

می‌تواند کارت مشخص شده را از میان کارتهای موجود در deck خود حذف کند.

خطاها:

۱. اگر اسم کارت اشتباه بود باید خطای زیر را نمایش دهید.

card name is invalid

۲. اگر چنین کارتی در میان کارتهایی که عضو deck شده‌اند وجود نداشت، باید عبارت زیر را چاپ کنید.

you don't have this type of card in your deck

در صورتی که با خطایی روبه‌رو نشدیم، کارت با موفقیت از دست فرد حذف می‌شود و پیام زیر چاپ می‌گردد.

card <card name> unequipped from your deck successfully

- برای نشان دادن دست، می‌توان از دستور

show my deck

استفاده کرد که با استفاد از آن می‌توان در قالب زیر، کارتهای موجود در آن را مشاهده کرد.

1.<card name>
2.<card name>
...

- توجه کنید که تمام کاراکترهای خروجی lowercase هستند.
- نشان دادن رتبه کاربر با استفاده از دستور زیر انجام می‌گردد.

show my rank

توجه کنید که کاربرها بر اساس experience ای که دارند رتبه‌بندی می‌شوند و اگر experience دو کاربر باهم برابر بود، کاربرها به ترتیب الفبا نمایش داده می‌شوند.

your rank:<rank>

- با استفاده از دستور

show ranking

می‌توان رده‌بندی کلی را مشاهده کرد که به فرمت زیر است:

1.username:<username> experience:<experience>
2.username:<username> experience:<experience>
...

منوی فروشگاه:

در این بخش کاربر می‌تواند به خرید و فروش کارتهای مختلف بپردازد و برای خرید هر نوع از کارتها تعداد مشخصی سکه بپردازد و یا کارتهایش را بفروشد و تعدادی سکه جدید کسب نماید. ارزش هر کدام از کارتهای بازی به صورت زیر است:

Pokemon	Value(coin)
Dragonite	10
Tepig	13
Lugia	11
Ducklett	15
Pineco	9
Rowlet	12
Pink (energy)	5
Yellow (energy)	5

همچنین برای فروش کارت‌ها نیز ارزش‌شان به صورت زیر است:

Pokemon	Value(coin)
Dragonite	8
Tepig	10
Lugia	9
Ducklett	11
Pineco	7
Rowlet	9
Pink (energy)	3
Yellow (energy)	3

- با استفاده از دستور زیر

buy card <card name>

می‌توان یک کارت جدید را خریداری کرد و به کارتهایی که کاربر دارد، اضافه کرد.

خطاها:

- اگر موجودی حساب برای خرید یک کارت کافی نبود، عبارت

not enough coin to buy <card name>

را باید چاپ کنید.

- اگر اسم کارت اشتباه بود و چنین کارتی وجود نداشت، عبارت

card name is invalid

را باید چاپ کنید.

در صورت موفقیت آمیز بودن خرید، پیام زیر چاپ می‌شود.

card <card name> bought successfully

- با استفاده از دستور

sell card <card name>

می‌توانیم یکی از کارتهایی که داریم را بفروشیم. دقت کنید که هم در بخش خرید و هم در بخش فروش، منظور از اسم کارت همان pink، dragonite و ... است و منظور fire و ... نیست.

دقت کنید کارتهایی که حذف می‌شوند از ابتدای لیست کارتها حذف می‌شوند.

خطاها:

- اگر اسم کارت اشتباه بود و چنین کارتی وجود نداشت، عبارت زیر را باید چاپ کنید.

card name is invalid

- اگر فرد کارت ذکرشده را برای فروش نداشت، عبارت زیر را چاپ کنید.

you don't have this type of card for sell

در صورت موفقیت آمیز بودن فروش کارت، پیام زیر چاپ می‌شود.

card <card name> sold successfully

قوانین بازی:

بر روی میز بازی هر بازیکن ۱ پوکمان فعال داشته که فقط همین کارت می‌تواند از اکشن خود استفاده کند. همچنین هر بازیکن می‌تواند حداکثر ۳ پوکمان در بخش bench خود داشته باشد که حین بازی قابلیت تعویض با پوکمان active دارند. بقیه‌ی کارتها هم در دست بازیکن قرار دارند که در ابتدای بازی ۱۲ تا هستند.

دسته‌بندی کارتها:

در بازی ۲ نوع کارت مختلف داریم؛ کارتهای پوکمان و کارتهای انرژی. پوکمان‌ها بر روی جایگاه‌های میز (bench و active) قرار می‌گیرند و کارتهای انرژی با قرار گرفتن در زیر کارتهای پوکمان به قدرت آنها می‌افزایند.

کارتهای پوکمان:

در این بازی ۳ نوع اصلی کارت پوکمان آتش، گیاه و آب داریم که هر پوکمان در یکی از این ۳ کلاس قرار می‌گیرد. هر کدام از کلاس‌های پوکمان یک قابلیت passive مخصوص خود دارند که به طور خودکار اعمال می‌شود:

کلاس پوکمان	قابلیت passive
آتش	اهداف مورد تهاجمشان در نوبت بعدی می‌سوزند
آب	در برابر سوختن مقاومند. (فعال یا ذخیره بودن اهمیتی ندارد.)
گیاه	یک سپر دارد که در صورت فعال بودن پوکمان عمل می‌کند. به این صورت که اول هر نوبت دوباره پر می‌شود و به اندازه‌ی ۱۵ واحد hitpoint دارد.

- توجه کنید که وقتی می‌گوییم نوبت بعدی یعنی پس از حمله‌ی ما، وقتی نوبت دشمن می‌شود آن پوکمان خاصیت burning دارد و در انتهای نوبت دشمن ۱۰ عدد از hitpoint او کم می‌شود.
- اگر پوکمان کلاس آتش از اکشن خود استفاده نکند، قاعدتا passive آن بر روی دشمنان هم اعمال نمی‌شود.
- در ادامه درباره‌ی سوختن توضیح داده خواهد شد.
- وقتی در دوری از بازی یک گیاه فعال بود (بنابراین ۱۵ واحد سپر داشت) اگر damage نهایی حاصل از فرمول ذکر شده در پایین برای مثال ۱۲ بود، چیزی از hitpoint گیاه کم نمی‌شود و اگر ۱۷ بود، ۲ واحد از hitpoint گیاه کم می‌شود.

پوکمان‌ها علاوه بر قابلیت passive مخصوص کلاس خود، یک action منحصر به فرد دارند:

پوکمان Dragonite (آتش):

پوکمان فعال دشمن را مورد حمله قرار می‌دهد.



پوکمان Tepig (آتش):



پوکمان فعال دشمن را با تمام power خود مورد حمله قرار می‌دهد و به تمام پوکمان‌های bench حریف با $0.2 * \text{قدرت اصلی}$ (که همان ۵ می‌شود) که باید در فرمول گفته شده برای محاسبه‌ی damage جایگذاری کنید، دمیج وارد می‌کند.

$\text{activeDamage} = \text{power} * \text{energy1} * \text{energy2} * \text{weakness} * \text{resistance}$
 $\text{benchDamage} = (0.2 * \text{power}) * \text{energy1} * \text{energy2} * \text{weakness} * \text{resistance}$

• بر روی پوکمان‌های bench نیز passive کلاس fire اعمال می‌شود.

پوکمان Lugia (آب):

پوکمان فعال دشمن را مورد حمله قرار می‌دهد و برای نوبت بعدی آن را به خواب می‌برد.



پوکمان Ducklett (آب):

پوکمان انتخاب شده (فعال یا ذخیره) حریف را مورد حمله قرار داده و آن را معلول می‌کند.



پوکمان Pineco (گیاه):

خود را heal می‌کند.



پوکمان Rowlet (گیاه):

پوکمان انتخاب شده میان پوکمان‌های ذخیره خودی را heal می‌کند.



- توجه کنید که در هر جای بازی برای اشاره به پوکمان‌ها در card name، تماماً lowercase می‌نویسیم.

ویژگی‌های power و hitpoint:

هر پوکمان یک عدد hitpoint دارد که بیان‌گر مقدار سلامتی آن پوکمان است. هر پوکمان اول بازی به اندازه‌ی حداکثر hitpoint خود سلامتی دارد.

- در نظر داشته باشید با heal شدن hitpoint آن پوکمان بیشتر از مقدار max hitpoint نمی‌تواند بشود.

- در آخر نوبت هر بازیکن، damage و heal‌های اعمال شده همگی اعمال می‌شوند و در نهایت اگر hitpoint کارتی کوچکتر یا مساوی صفر بود می‌میرد و از بازی حذف می‌شود.

همچنین هر پوکمان یک عدد power دارد که میزان تغییرات action آن پوکمان بر روی هدف خود (heal یا attack) را مشخص می‌کند.

پوکمان	Max Hitpoint	Power
Dragonite	120	40
Tepig	140	25
Lugia	90	20
Ducklett	70	20
Pineco	110	25
Rowlet	180	40

ضعف و مقاومت در کارت‌ها:

هر کارت پوکمان ممکن است نسبت به یک کلاس پوکمان ضعف داشته باشد به این معنا که از آن کلاس آسیب‌پذیرتر باشد. دمیج و heal‌هایی که از طرف یک کارت به کارت دیگر اعمال می‌شود، مقدار نهایی آن نیست و بر اساس ضعف و مقاومت کارت مورد هدف، مقدار نهایی آن محاسبه می‌شود. محاسبه به این صورت است که اگر برای مثال کارتی نسبت به کلاس آتش با ضریب ۲ ضعف داشت، حمله کارت‌های آتش روی آن ۲ برابر تاثیر می‌گذارد.

- توجه داشته باشید که اگر ضعف در مقابل کلاس آتش بود، بر روی اثر سوختن تاثیر ندارد.
- اگر ضعف در برابر کلاس گیاه بود، میزان تاثیرگذاری heal بر روی آن پوکمان تغییر می‌کند.

ضعف و مقاومت کارت‌ها در برابر یکدیگر، در جدول زیر آمده‌است.

پوکمان/کلاس مهاجم	آتش	آب	گیاه
Dragonite	-	1.2	-

پوکمان/کلاس مهاجم	آتش	آب	گیاه
Tepig	-	2	1.3
Lugia	1.3	-	-
Ducklett	-	-	1.5
Pineco	-	-	-
Rowlet	1.3	-	-

همچنین هر پوکمان ویژگی مقاومت دارد که به طور کلی میزان تاثیرگذاری هر اکشن (دمیج یا هیل) را بر روی آن‌ها مشخص می‌کند.

پوکمان	مقاومت
Dragonite	0.7
Tepig	0.8
Lugia	0.7
Ducklett	0.6
Pineco	0.9
Rowlet	0.5

شرایط خاص پوکمان‌ها:

پوکمان‌ها ممکن است دچار ۳ حالت خاص شوند که به توضیح آن‌ها می‌پردازیم.

خواب:

اگر یک پوکمان در یک نوبت در حالت خواب قرار گرفته باشد، در آن راند نمی‌تواند اکشنی انجام دهد و همچنین نمی‌تواند تعویض شود. (مکانیزم تعویض در ادامه توضیح داده خواهد شد.)

سوختن:

اگر یک پوکمان در یک راند در حال سوختن باشد، در انتهای نوبت ۱۰ واحد از hitpoint او کم می‌شود.

- بر ۱۰ واحد آسیب وارد شده بر اثر سوختن، هیچ عاملی مثل مقاومت، انرژی، weakness و... به جز سپر کلاس گیاهان تاثیرگذار نیست و hitpoint به‌طور صریح ۱۰ واحد کم می‌شود.

معلولیت:

اگر یک پوکمان در حالت معلولیت قرار گیرد، انرژی‌های نسبت داده شده به آن از بین می‌روند. این حالت بر خلاف سوختن و خواب، اثرش بر روی پوکمان نمی‌ماند و به صورت آنی عمل می‌کند. به این معنا که بلافاصله در لحظه‌ی اعمال شدن این اثر، کارتهای انرژی پوکمان هدف از بین می‌روند.

کارتهای انرژی:

کارتهای انرژی، کارتهایی هستند که به کارتهای پوکمان کمک می‌کنند و power آنها را در action بعدی تحت تاثیر قرار می‌دهند. به عبارتی قدرت حمله را در کارتهایی از نوع آتش و آب زیاد می‌کنند و همچنین باعث افزایش قدرت heal در گیاه می‌شوند که قوانین آنها از جدول زیر پیروی می‌کند. برای مثال اگر قدرت (power) یک نوع پوکمان خاص ۱۰۰ باشد و این عدد ۱.۲ باشد، قدرت در آن حمله به ۱۲۰ می‌رسد.

- در هر جای بازی برای اشاره به انرژی زرد و صورتی در card name به ترتیب عبارات زیر را می‌نویسیم:

pink
yellow

Yellow	Pink	Energy Type
1	1	Fire

Yellow	Pink	Energy Type
1.2	1.05	Water
1.2	1.15	Plant

در نهایت مقدار damage وارد شده که باید از hitpoint کارت فعال دشمن کم شود، از رابطه زیر به دست می‌آید:

$$damage = -1 \times power \times (energy_1) \times (energy_2) \times (weakness) \times (resistance)$$

همچنین برای heal نیز مثل damage، مقدار زیاد شدن hitpoint از رابطه زیر بدست می‌آید و مقاومت پوکمان مقصد روی آن تاثیر ندارد:

$$heal = power \times (energy_1) \times (energy_2) \times (weakness) \times (resistance)$$

- به عبارتی ضعف در برابر گیاه باعث heal شدن بیشتر می‌شود.

به نکات زیر در رابطه با آن توجه کنید:

۱. به جز بخش power و resistance که اجباری هستند، بقیه‌ی بخش‌ها ممکن است نباشند.
۲. بخش weakness همان ضعف و قدرت در برابر کارتهاست که اگر یک کارت ضعف ۲ برابری در مقابل کارت دیگری داشت باید در ۲ ضرب شود.

- برای مثال فرض کنید کارت dragonite با ۲ کارت انرژی زرد در زیر خود اکشن خود را انجام می‌دهد و به کارت اکتیو حریف (rowlet با hitpoint = 100) حمله می‌کند. محاسبه‌ی آسیب وارد شده به شکل زیر خواهد بود:

$$damage = -1 \times power \times (energy_1) \times (energy_2) \times (weakness) \times (resistance) = -1 \times 40 \times 1 \times 1 \times 1.3 \times 0.5 = -26$$

و همچنین پوکمان rowlet از کلاس گیاه است و هر دست به اندازه 15 واحد shield دارد. پس در نهایت hitpoint نهایی rowlet برابر است با:

$$hitpoint = 100 - (26 - 15) = 89$$

و در دست بعدی به خاطر passive کلاس آتش، پوکمان rowlet می‌سوزد و علاوه بر هر تغییر دیگر بر hitpoint آن پوکمان، ۱۰ واحد نیز کم‌تر می‌شود.

یعنی اگر برای مثال در دست بعدی توسط یک اکشن تغییرات hitpoint این پوکمان برابر با ۲۰ محاسبه شد، ۱۰ واحد هم آتش می‌گیرد و ۱۵ واحد سپر دارد پس ۱۵ واحد در مجموع از آن کم می‌شود.

اگر هم برای مثال در دست بعدی هیچ تهاجمی به آن نشود فقط ۱۰ واحد به دلیل سوختن باید از آن کم شود که به دلیل ۱۵ واحد سپری که دارد، تغییری نمی‌کند.

به نکات زیر در رابطه با کارت های انرژی توجه کنید:

- پس از انجام action، کارت انرژی از بین می‌رود.
- برای استفاده کردن از یک کارت انرژی آن کارت باید در زیر یک پوکمان قرار بگیرد تا بر اساس مقدارهای موجود در جدول بالا به قدرت آن کمک کند.
- وقتی یک کارت انرژی در زیر یک پوکمان قرار داده می‌شود، با آن پوکمان جابه‌جا می‌شود و تا زمان استفاده شدن از انرژی یا مرگ آن پوکمان در زیر پوکمان می‌ماند.
- زیر هر پوکمان حداکثر ۲ کارت انرژی صرف نظر از صورتی یا زرد بودنشان قرار می‌گیرد. اگر دو کارت انرژی زیر پوکمان باشد، هنگام action هر دو بر power تاثیر می‌گذارند و کارت انرژی قدیمی‌تر از بین می‌رود. برای مثال اگر ابتدا کارت زرد سپس صورتی را زیر یک پوکمان قرار دهیم و action را انجام دهیم، هر دو کارت انرژی تاثیر خود را می‌گذارند و بعد از آن کارت زرد از بین می‌رود.
- زیر یک پوکمان ممکن است همزمان کارت انرژی زرد و صورتی قرار گیرند.

روند بازی:

میز بازی برای هر فرد به صورت زیر است:

- هر فرد يك دست دارد كه حريف قادر به مشاهده‌ی محتويات آن نيست. در ابتدای بازی ۱۲ عدد كارت در دست هر بازیكن هست كه آنها را برای خود از طريق بخش پروفایل منو انتخاب كرده است.
- كارت‌هایی كه در حین بازی از دست بازیكن با دستور put card بر روی میز بازی گذاشته می‌شوند، از deck او حذف می‌شوند و برای بازی بعدی باید در صورت نیاز خریداری شده و در profile menu، برای بازی بعدی equip شوند. به عبارتی كارت‌ها چه در بازی از بین بروند چه تا آخر بازی سالم بمانند، از deck حذف شده و يك بار مصرف هستند.
- هر فرد ۳ جایگاه bench دارد كه حداكثر يك پوكمان در هر كدام از آنها جای می‌گیرد. برای مشخص كردن این ۳ جایگاه از اعداد ۱ تا ۳ استفاده می‌كنیم.
- هر فرد يك جایگاه active یا فعال دارد كه يك پوكمان در آن قرار دارد. برای مشخص كردن جایگاه فعال از عدد ۰ استفاده می‌كنیم.
- اگر در انتهای نوبت يك بازیكن جایگاه پوكمان فعال خالی بود، آن بازیكن بازنده‌ی بازی است. (توضیحات بیشتر در بخش برد و باخت داده شده است.)
- بنابراین هر بازیكن باید مراقب باشد كه پس از مرگ پوكمان active خود يك كارت از دست خود یا يك كارت از bench را در در جایگاه active قرار بدهد.
- جایگاه bench و active هر دو بازیكن برای یكدیگر قابل رویت است.
- اولین نوبت با بازیكنی است كه بازی را شروع كرده است.

عملیات‌ها در يك نوبت

زمانی كه نوبت يك بازیكن می‌شود، تعدادی عملیات می‌توانند انجام دهند و در نهایت با انجام دادن اكشن پوكمان فعال به نوبت خود پایان دهند. البته می‌توان بدون انجام دادن اكشن نهایی نیز اتمام نوبت خود را اعلام كرد.

مشاهده وضعیت میز

با دستور زیر وضعیت كل میز نشان داده می‌شود.

```
show table
```

که باید کل بازی به غیر از دست حریف را به بازیکن به این صورت نمایش دهد:

round <round number>

your active card:

<pokemon name>|<energy name>|<energy name>

your hand:

1.<card name>

2.<card name>

...

k.<card name>

your bench:

1.<pokemon name>|<energy name>|<energy name>

2.<pokemon name>|<energy name>|<energy name>

3.<pokemon name>|<energy name>|<energy name>

<enemy username>'s active card:

<pokemon name>|<energy name>|<energy name>

<enemy username>'s bench:

1.<pokemon name>|<energy name>|<energy name>

2.<pokemon name>|<energy name>|<energy name>

3.<pokemon name>|<energy name>|<energy name>

- شماره راند از ۱ شروع شده و هر بار که نوبت شروع کننده بازی می‌شود ۱ عدد افزایش پیدا می‌کند. توجه کنید که در توضیحات بازی هر جایی که از عبارت "راند" یا "نوبت" یا "دور" صحبت شده است، منظور نوبت یک بازیکن است. تنها در این دستور است که round number را برابر با تعداد راند سپری شده تقسیم بر ۲ نمایش می‌دهیم.
- ترتیب نمایش دست بازیکن دقیقاً مشابه دستور show my deck باید به همان ترتیب equip شدن کارتها در منوی پروفایل باشد.
- اگر k کارت در دست موجود بود، تا شماره‌ی kام نمایش داده می‌شود.
- اگر یک جایگاه bench خالی بود، در آن خط بعد از شماره‌ی خط چیزی چاپ نمی‌شود و جلوی کاراکتر «.» خالی می‌ماند و در خط بعدی ادامه می‌دهیم.

- اگر جایگاه active خالی بود یک خط خالی چاپ کنید.
- اگر انرژی زیر کارتی وجود نداشت، میان «||» چیزی چاپ نمی‌شود. (خطوط جدا کننده همیشه چاپ می‌شوند).
- همه‌ی عبارات در lowercase چاپ می‌شوند.

مشاهده وضعیت کارت

با دو دستور دستور زیر وضعیت جایگاه مشخص شده برای بازیکن حریف یا برای خود نشان داده می‌شود.

show my info <place number>

show enemy info <place number>

خطاها:

۱. اگر عدد وارد شده نامعتبر بود، خطای زیر را چاپ کنید.

invalid place number

۲. اگر کارتی در مکان مشخص شده وجود نداشت، عبارت زیر را چاپ کنید.

no pokemon in the selected place

اگر مشکلی نبود، اطلاعات آن خانه را به شکل زیر نشان دهید:

pokemon: <pokemon name>

special condition: <burning/sleep>

hitpoint: <current hitpoint>/<max hitpoint>

energy 1: <first energy>

energy 2: <second energy>

- همه‌ی خطها چاپ می‌شوند و اگر انرژی یا حالت خاصی موجود نبود جلوی : خالی می‌ماند.

- توجه داشته باشید که حالت معلولیت (paralyzed) بر روی پوکمانی نمی‌ماند و صرفاً بعد از اعمال شدن، کارتهای انرژی کارت هدف را از بین می‌برد. به همین دلیل فقط حالت خواب و سوختن بر روی پوکمان می‌تواند ماندگار باشد.
- ترتیب چاپ انرژی‌ها مشابه با دستور قبلی است.
- همه‌ی عبارات در lowercase چاپ می‌شوند.
- مقادیر max hitpoint و current hitpoint با ۲ رقم اعشار نمایش داده شوند.

بازی کردن کارتی از دست

با دستور زیر، کارت انتخابی را از دست خود در جایگاه انتخابی قرار می‌دهیم. (۰ برای جایگاه active و ۱، ۲ و ۳ برای bench)

```
put card <card name> to <place number>
```

خطاها:

۱. اگر اسم کارت وارد شده اشتباه بود و کارت با چنین نامی وجود نداشت، عبارت زیر را چاپ کنید.

```
card name is invalid
```

۲. اگر از اسم کارتی که وارد شده نداشتیم، باید عبارت زیر را چاپ کنید.

```
you don't have the selected card
```

۳. اگر عدد مقصد وارد شده نامعتبر بود، خطای زیر را چاپ کنید.

```
invalid place number
```

۴. اگر کارت انتخابی پوکمان باشد و جایگاه مد نظر پر بود، عبارت زیر چاپ شود.

```
a pokemon already exists there
```

۵. اگر کارت انتخابی انرژی باشد و در جایگاه انتخابی پوکمانی وجود نداشت، عبارت زیر چاپ شود.

no pokemon in the selected place

۶. اگر کارت انتخابی انرژی باشد و پوکمان انتخاب شده از قبل ۲ کارت انرژی داشته باشد، عبارت زیر چاپ شود.

pokemon already has 2 energies

۷. اگر کارت انتخابی انرژی باشد و در این دست یکبار کارت انرژی را با این دستور اضافه کرده باشد باید عبارت زیر چاپ شود. (دقت کنید بازیکن در هر دست حداکثر یکبار می‌تواند این دستور را برای کارت انرژی به کار ببرد ولی بازی کردن پوکمان‌ها در یک دست محدودیت ندارد).

you have already played an energy card in this turn

پس از این دستور در صورت عدم وقوع خطا، فرد کارت انتخاب شده را در ناحیه‌ی تعیین شده می‌گذارد و عبارت زیر چاپ می‌شود.

card put successful

تعویض کارت فعال

بازیکن در ابتدای هر نوبت خود می‌تواند کارت فعال خود را با یک کارت موجود در bench جابه‌جا کند.

- توجه کنید اگر کارت ما دارای شرط خاص خواب باشد نمی‌تواند تعویض شود و همچنین اگر یک کارت تعویض شده دارای شرایط خاص مثل معلولیت یا سوختن باشد کماکان شرایط ویژه را تا زمانی که برای آن مقدر شده است پس از تعویض نیز دارد برای مثال اگر در حال سوختن باشد دوباره در انتهای نوبت خود ۱۰ واحد از hitpoint او کم می‌شود.

برای جابه‌جایی کارت فعال با یکی از کارت‌های موجود در نیمکت می‌توانیم از دستور زیر استفاده کنیم.

substitute active card with bench <bench number>

که number می تواند یکی از اعداد ۱ تا ۳ باشد که نشان دهنده‌ی این است که با کدام پوکمان روی bench تعویض می‌شود. توجه کنید که اگر دو پوکمان باهم تعویض شدند، کارتهای انرژی همراه آنها که زیرشان است نیز تعویض می‌شوند.

خطاها:

۱. اگر عدد وارد شده نامعتبر بود خطای زیر را چاپ کنید.

invalid bench number

۲. اگر در جایگاه انتخابی روی نیمکت پوکمانی وجود نداشت، عبارت زیر چاپ شود.

no pokemon in the selected place

- توجه داشته باشید که اگر جایگاه اکتیو خالی بود مشکلی نیست و صرفاً کارت bench و انرژی‌هایش به جایگاه active جابه‌جا می‌شوند.

۳. اگر پوکمان active در حالت خواب قرار داشته باشد نمی‌تواند تعویض شود. پس عبارت زیر را چاپ کنید.

active pokemon is sleeping

اگر هیچ خطایی رخ نداد عملیات را انجام داده و عبارت زیر را چاپ کنید.

substitution successful

اعلام پایان نوبت

اگر بازیکن قصد دارد بدون انجام اکشن کارت فعال خود به نوبت خود خاتمه دهد از دستور زیر استفاده می‌کند.

end turn همه‌ی تغییرات اعمال می‌گردند و در صورت ادامه‌ی بازی، نوبت به حریف داده می‌شود و دستور زیر چاپ می‌شود.

<username>'s turn که username همان یوزرنیم بازیکنی است که نوبتش شده است.

انجام اکشن

با دستور زیر، پوکمان فعال اکشن خود را انجام می‌دهد و نوبت به حریف داده می‌شود. دو پوکمان Ducklett و Rowlet نیاز به تعیین هدف دارند که پارامتر target را با اعداد مخصوص جایگاه هدف پر می‌کنند و بقیه‌ی پوکمان‌ها نیازی به آن ندارند.

```
execute action -t <target>
```

خطاها:

۱. اگر پوکمانی فعال نبود، عبارت زیر چاپ شود.

```
no active pokemon
```

۲. اگر نوع پوکمان فعال با بود یا نبود پارامتر t- مغایرت داشت، عبارت زیر چاپ شود.

```
invalid action
```

۳. اگر عدد target اعلام شده برای Ducklett از ۰ تا ۳ نبود و یا برای Rowlet از ۱ تا ۳ نبود، عبارت زیر چاپ شود.

```
invalid target number
```

۴. اگر در مقصد مشخص شده پوکمانی وجود نداشت عبارت زیر چاپ شود.

```
no pokemon in the selected place
```

۵. اگر پوکمان فعال در حالت خواب قرار داشته باشد، این راند نمی‌تواند از action خود استفاده کند. پس عبارت زیر چاپ شود.

```
active pokemon is sleeping
```

در نهایت در صورت عدم وقوع هیچ اروری، نوبت بازیکن فعلی تمام می‌شود و در صورت ادامه‌ی بازی نوبت به بازیکن حریف داده می‌شود و عبارت زیر را چاپ کنید.

```
action executed successfully
<username>'s turn
```

اتمام نوبت

پس از اتمام نوبت هر بازیکن، تغییرات hitpoint بر اساس فرمول گفته شده محاسبه می‌شود و اگر hitpoint پوکمانی بر روی میز از صفر بیشتر نبود آن پوکمان و همه‌ی انرژی‌های زیرش از بین می‌روند و دست بعدی شروع می‌شود.

- همه‌ی اثرات heal و damage اعمال شده و اگر hitpoint جدید از سقف بیشتر بود، برابر با سقف می‌شود.

برد و باخت

در آخر هر نوبت هر بازیکن اگر آن بازیکن پوکمان active نداشته باشد بازنده‌ی بازی است. یعنی بازیکن در صورت مرگ پوکمان اکتیو باید مراقب باشد که تا وقتی که در دست خود یا bench خود پوکمان دارد در اتمام نوبتش یک پوکمان را (از bench با دستور substitute یا از دست با دستور put card) در جایگاه active قرار داده باشد که بازنده نشود. وقتی دیگر پوکمانی برای خود ندارد چاره‌ای ندارد و می‌بازد. پس هدف برد نابود کردن همه‌ی پوکمان‌های حریف است.

اتمام بازی

در انتهای بازی experience هر فرد بر اساس تعداد کارتهایی از دشمن که نابود شده‌اند اضافه می‌شود:

$$experience = experience + (kills \times 10)$$

$$coin = coin + [(reduce \div 10)]$$

- منظور از بخش reduce میزان تغییرات hitpoint کارتهای دشمن نسبت به ابتدای بازی است و اگر یک کارت از بین برود این مقدار برای آن کارت منطقاً برابر با max hitpoint آن کارت می شود؛ پس اتفاقات رخ داده در حین بازی به جز از بین رفتن یک کارت در میزان reduce موثر نیست و تنها hitpoint ابتدای بازی و انتهای آن مهم است. (دقت کنید که جزصحيح مقدار تقسیم شده coin به آن اضافه می شود).
- منظور از kills، تعداد کارتهای نابود شده از حریف است؛ برای مثال اگر یک پوکمان با ۲ کارت انرژی را بکشیم، ۳ واحد به آن اضافه می شود. یا اگر یک پوکمان با ۲ کارت انرژی را معلول کنیم، ۲ واحد به kills اضافه می شود. حتی اگر حریف خودش از یک کارت انرژی استفاده کند نیز یک عدد به kills ما اضافه می شود.
- هم بازنده و هم برنده به experience و coinشان افزوده می شود.

پس از اتمام بازی، برنده ی بازی در قالب زیر نمایش داده می شود و دوباره به منوی اصلی یا همان main menu باز می گردیم.

Winner: <winner's username>

تست نمونه

می توانید دو مورد تست نمونه را از اینجا دانلود بکنید.

بدون مقدمه

بدون مقدمه، در این سوال، قرار است یک فایل منیجر ساده را پیاده سازی کنیم.

فاز صفر

در این بخش به فعالیت های کاربری میپردازیم.

ایجاد کاربر

با دستور زیر یک کاربر با نام کاربری و رمز عبور مشخص شده می‌سازیم.

```
create_user -username <username> -password <password>
```

- اگر کاربری با نام کاربری مذکور وجود داشت، خطای زیر را چاپ کنید.

```
user already exists
```

- اگر نام کاربری شامل کاراکتری غیر از حروف الفبای انگلیسی و اعداد 0 تا 9 و کاراکتر _ بود، خطای زیر را چاپ کنید.

```
invalid username format
```

- رمز عبور باید شامل یک حرف Upper case، یک حرف lower case و یک عدد باشد و حداقل شامل 8 کاراکتر باشد. اگر رمز عبور وارد شده یکی از این ویژگی ها را نداشت، خطای زیر را چاپ کنید.

```
invalid password
```

- با اضافه کردن تگ دلخواه one_time می‌توان اکانتی ساخت که یک بار مصرف است و بعد از logout کردن، حساب کاربری از بین می‌رود. مثال:

```
create_user -username <username> -password <password> -one_time
```

- در صورتی که خطایی وجود نداشت پیام زیر را چاپ کنید و سپس کاربر می‌تواند برای لاگین کردن اقدام کند.

```
register successful
```

لاگین

با دستور زیر، کاربر می‌تواند به حساب کاربری خود وارد شود.

```
login -username <username> -password <password>
```

- اگر نام کاربری موجود نبود، خطای زیر را چاپ کنید.

```
user doesn't exist
```

- اگر پسورد ارائه شده با نام کاربری مذکور مطابقت نداشت، خطای زیر را چاپ کنید.

```
password doesn't match
```

- در صورتی که خطایی وجود نداشت پیام زیر را چاپ کنید و سپس کاربر می‌تواند به حساب کاربری خود وارد شود.

```
login successful
```

پس از ورود به حساب کاربری، یک کاربر می‌تواند فعالیت های زیر را انجام دهد.

اشتراک گذاری یک دایرکتوری

با دستور زیر، یک دایرکتوری از میان دایرکتوری های موجود را با نام مشخص شده، با کاربر مشخص شده به اشتراک میگذاریم.

```
share -dir <dir_name> -target <username>
```

- اگر کاربر دایرکتوری ای با نام مشخص شده نداشت، خطای زیر را چاپ کنید.

```
no such file or directory
```

- اگر کاربر مقصد موجود نبود، خطای زیر را چاپ کنید.

```
no such user exists
```

- اگر دایرکتوری پیش تر با کاربر به اشتراک گذاشته شده بود، خطای زیر را چاپ کنید.

```
already shared
```

- اگر نام دایرکتوری، root بود، خطای زیر را نمایش دهید.

```
you can't share root
```

- در صورتی که خطایی وجود نداشت پیام زیر را چاپ کنید.

```
shared successfully
```

اشتراک گذاری یک فایل

با دستور زیر، یک فایل از میان فایل های موجود را با نام مشخص شده، با کاربر مشخص شده به اشتراک میگذاریم.

```
share -file <file_name> -target <username>
```

- اگر کاربر فایلی با نام مشخص شده نداشت، خطای زیر را چاپ کنید.

```
no such file or directory
```

- اگر کاربر مقصد موجود نبود، خطای زیر را چاپ کنید.

```
no such user exists
```

- اگر فایل پیش‌تر با کاربر به اشتراک گذاشته شده بود، خطای زیر را چاپ کنید.

```
already shared
```

- در صورتی که خطایی وجود نداشت پیام زیر را چاپ کنید.

```
shared successfully
```

- برای هر دو مورد بالا، اگر کاربر در دسترس نبود (یعنی کاربر مذکور، کاربر فعلی را بلاک کرده)، پیام زیر را چاپ کنید.

```
can't share, you are blocked
```

نمایش فایل ها و دایرکتوری های اشتراک گذاشته شده

با دستور زیر، ابتدا تمام فولدر ها و سپس تمام فایل ها با فرمت عادی و سپس فایل های زیبای نمایش داده می‌شوند که ما با کاربر مشخص شده به اشتراک گذاشته ایم و سپس تمام فولدر ها و فایل های عادی و فایل های زیبای که کاربر مشخص شده با ما به اشتراک گذاشته شده، نمایش داده می‌شود. (نمایش به ترتیب زمان اشتراک گذاری می‌باشد).

- نکته: این دستور، حتی در زمانی که کاربر مذکور بلاک شده باشد نیز، معتبر است.

```
view -shared -username <username>
```

مثال:

```
shared to <username> by me:
```

```
dir:
```

```
1. dir_name1
```

```
2. dir_name2
```

```
...
```

```
file:
```

```
1.file_name1
```

```
2. file_name2
```

```
...
```

```
zip:
```

```
1.file_name1
```

```
2. file_name2
```

```
...
```

```
shared to me by <username>:
```

```
dir:
```

```
1. dir_name1
```

```
2. dir_name2
```

```
...
```

```
file:
```

```
1.file_name1
```

```
2. file_name2
```

```
...
```

```
zip:
```

```
1.file_name1
```

```
2. file_name2
```

```
...
```

- اگر کاربر مقصد موجود نبود، خطای زیر را چاپ کنید.

```
no such user exists
```

- اگر موردی برای نمایش موجود نبود، عبارت زیر را چاپ کنید.

```
share list empty
```

- اگر تگ دلخواه me در کامند موجود بود، تنها مواردی که خود کاربر با کاربر مذکور به اشتراک گذاشته نمایش داده شود.
- اگر تگ دلخواه user در کامند موجود بود، تنها مواردی که کاربر مذکور با خود کاربر به اشتراک گذاشته نمایش داده شود.

```
view -shared -username <username> -(me/user)
```

- در هر دو مورد، اگر لیست مذکور خالی بود عبارت زیر را چاپ کنید.

```
share list empty
```

اشتراک گذاری با همه

با دستور زیر، می‌توان دایرکتوری مورد نظر را با تمامی کاربران در دسترس به اشتراک گذاشت. (کاربرانی که بلاک شده اند یا ما را بلاک کردند یا پیشتر این دایرکتوری را دریافت کرده اند، آن را دریافت نمی‌کنند. همچنین نام دایرکتوری، به فرمت ذکر شده در انتهای همین فاز می‌باشد.)

```
share_all -dir <dir_name>
```

- اگر کاربر دایرکتوری ای با نام مورد نظر نداشت، خطای زیر را چاپ کنید.

```
no such file or directory
```

- اگر نام دایرکتوری، root بود، خطای زیر را نمایش دهید.

```
you can't share root
```

با دستور زیر، می‌توان فایل مورد نظر را با تمامی کاربران در دسترس به اشتراک گذاشت. (کاربرانی که بلاک شده اند یا ما را بلاک کردند یا پیشتر این فایل را دریافت کرده اند، آن را دریافت نمی‌کنند. همچنین نام فایل، به فرمت ذکر شده در انتهای همین فاز می‌باشد.)

```
share_all -file <file_name>
```

- اگر کاربر فایلی با نام مورد نظر نداشت، خطای زیر را چاپ کنید.

```
no such file or directory
```

- برای هر دو مورد بالا، اگر کاربری که دایرکتوری یا فایل با او به اشتراک گذاشته نشده، در دسترس نبود (یعنی کاربری که کاربر فعلی را بلاک نکرده یا محتوا را دریافت نکرده، وجود نداشت)، پیام زیر را چاپ کنید.

```
no available user
```

- در صورتی که خطایی وجود نداشت پیام زیر را چاپ کنید.

```
shared successfully
```

بلاک کردن

با دستور زیر، دسترسی کاربر مشخص شده را برای اشتراک گذاری با خود، قطع می‌کنیم.

```
block -user <username>
```

- اگر کاربر مورد نظر وجود نداشت، خطای زیر را چاپ کنید.

```
no such user exists
```

- اگر کاربر مورد نظر از قبل بلاک شده بود، خطای زیر را چاپ کنید.

user already blocked

- در صورتی که خطایی وجود نداشت پیام زیر را چاپ کنید.

user <username> is now blocked

با دستور زیر، کاربر مشخص شده را از بلاک در می‌آوریم.

unblock -user <username>

- اگر کاربر مورد نظر وجود نداشت، خطای زیر را چاپ کنید.

no such user exists

- اگر کاربر مورد نظر، قبلاً بلاک نشده بود، خطای زیر را چاپ کنید.

user is not blocked

- در صورتی که خطایی وجود نداشت پیام زیر را چاپ کنید.

user <username> is now unblocked

با دستور زیر، ابتدا لیست تمامی افرادی که بلاک کرده اید نمایش داده می‌شود و سپس لیست تمامی افرادی که شما را بلاک کرده اند، نمایش داده می‌شود.

show blocklist

مثال:

```
blocked:
1. name1
2. name2
...
blocker:
1. name1
2. name2
...
```

- اگر لیست خالی بود، خط زیر را چاپ کنید.

```
block list empty
```

- نکته: در تمامی دستورات بالا، شامل اشتراک گذاری ها و بلاک و آنبلاک کردن، کاربر امکان وارد کردن username خود را ندارد. در صورت وارد کردن username خود، خطای زیر را چاپ کنید.

```
You can't interact with yourself!
```

ورود به منوی file manager

با دستور زیر، به منوی کنترل کردن دایرکتوری ها و فایل وارد می شویم. (فاز اول)

```
manage files
```

پس از این دستور، پیام زیر را چاپ کنید.

```
you are now in directory root
```

خروج

با دستور زیر از حساب کاربری خود خارج می شویم.

logout

پس از این دستور، پیام زیر را چاپ کند.

logged out successfully

و اگر حساب کاربری، یک بار مصرف بود، پیام زیر را نیز چاپ کنید.

account removed!

- در صورتی که وارد اکانتی نشده بودیم و این دستور وارد شد، خطای زیر را چاپ کنید.

There's no account to be logged out

با دستور زیر، هر جای برنامه که بودیم، برنامه terminate می‌شود.

exit

نکته:

اگر دستور وارد شده در این بخش، با موارد بالا همخوانی نداشت، خطای زیر را چاپ کنید.

invalid command

- نکته: نام دایرکتوری ها و فایل ها به صورت زیر است:

root/parent/dir_name

root/parent2/parent1/file_name.type

فاز یک

با ورود به این فاز، درون دایرکتوری root قرار می‌گیریم.

ساختن

با دستور زیر، در دایرکتوری کنونی، یک دایرکتوری با نام مشخص شده ساخته می‌شود.

```
create -dir <dir_name>
```

- اگر دایرکتوری ای با نام مشخص شده وجود داشت، خطای زیر را چاپ کنید.

already exists

- نام دایرکتوری می‌تواند شامل هر کاراکتری بجز اسپیس باشد. همچنین شروع نام تنها می‌تواند با حروف انگلیسی یا ارقام 0 تا 9 یا کاراکتر '.' باشد. در غیر اینصورت، خطای زیر را چاپ کنید.

incorrect name format

- اگر نام دایرکتوری با '.' شروع شود، آن فایل hidden خواهد بود.
- در صورتی که خطایی رخ نداد، پیام زیر را چاپ کنید.

directory created successfully!

با دستور زیر، در دایرکتوری کنونی، یک فایل با نام و فرمت مشخص شده ساخته می‌شود.

```
create -file <file_name> -format <format>
```

- فرمت های مدنظر، txt و mkv و mp3 و mp4 و csv و dat و exe و c و java می‌باشد.
- اگر فایلی با نام و فرمت ذکر شده موجود بود، خطای زیر را چاپ کنید.

already exists

- اگر فرمت مدنظر، در فرمت های ذکر شده موجود نبود، خطای زیر را چاپ کنید.

incorrect format

- نام دایرکتوری می‌تواند شامل هر کاراکتری بجز اسپیس باشد. همچنین شروع نام تنها می‌تواند با حروف انگلیسی یا ارقام 0 تا 9 یا کاراکتر '.' باشد. در غیر اینصورت، خطای زیر را چاپ کنید.

incorrect name format

- اگر نام فایلی با '.' شروع شود، آن فایل hidden خواهد بود.
- در صورتی که خطایی رخ نداد، پیام زیر را چاپ کنید.

file created successfully!

نمایش مخفی

با دستور زیر، ابتدا تمامی دایرکتوری ها و سپس تمامی فایل های مخفی به ترتیب زمان مخفی شدن نمایش داده می‌شوند.

```
show_hidden -password <user_password>
```

- اگر حساب کاربری یک بار مصرف بود، بدون وارد کردن تگ password نیز می‌تواند لیست را مشاهده کند.
- اگر پسورد ذکر شده با پسورد کاربر مطابقت نداشت، خطای زیر را چاپ کنید.

invalid password

- اگر مورد مخفی ای وجود نداشت، عبارت زیر را چاپ کنید.

no hidden item

نحوه نمایش لیست:

```
dir:
1. dir_name1
2. dir_name2
...
file:
1. file_name1
2. file_name2
...
zip:
1. zip_name1
2. zip_name2
...
```

انتقال

با دستور زیر، از دایرکتوری کنونی به دایرکتوری مذکور وارد می‌شویم.

```
cd -name <dir_name>
```

- اگر دایرکتوری مذکور وجود نداشت یا دسترسی به آن ممکن نبود (بیش از یک لایه فاصله یا مخفی) خطای زیر را چاپ کنید و سپس با توجه به دلیل خطا، خط دوم را چاپ کنید.

```
unable to open directory
no such directory (موجود نبودن)
hidden directory (مخفی بودن)
too deep (بیش از یک لایه فاصله)
```

- با دستور `cd ..` به دایرکتوری parent می‌رویم.
- با اضافه کردن فلگ `jmp`، می‌توان به دایرکتوری در هر جای سیستم رفت.

```
cd -name <dir_name> -jmp
```

- مجدداً اگر دایرکتوری نام برده موجود نبود، خطای زیر را چاپ کنید.

```
unable to open directory
no such directory
```

- اگر در دایرکتوری root، دستور cd .. وارد شد، خطای زیر را چاپ کنید.

```
root doesn't have parent node
```

در صورتی که خطایی رخ نداد، پیام زیر را چاپ کنید.

```
entered directory <directory_name>
```

تغییر نام

با دستور زیر، نام یک دایرکتوری درون دایرکتوری فعلی را تغییر می‌دهیم.

```
rename -dir <dir_name> -new <new_name>
```

- اگر نام جدید متفاوت از نام قبلی نبود، خطای زیر را چاپ کنید.

```
same name!
```

- اگر دایرکتوری موجود نبود، خطای زیر را چاپ کنید.

```
no such directory
```

- اگر نام جدید تکراری بود، خطای زیر را چاپ کنید.

```
name is already taken
```

با دستور زیر، نام یک فایل درون دایرکتوری فعلی را تغییر می‌دهیم.

```
rename -file <file_name> -new <new_name>
```

- اگر نام جدید متفاوت از نام قبلی نبود، خطای زیر را چاپ کنید.

same name!

- اگر فایل موجود نبود، خطای زیر را چاپ کنید.

no such file

- اگر نام جدید تکراری بود، خطای زیر را چاپ کنید.

name is already taken

- دقت کنید که اگر در ابتدای نام جدید، کاراکتر '.' وجود داشت، دایرکتوری یا فایل مذکور، مخفی می‌شود.

- در صورتی که خطایی رخ نداد، پیام زیر را چاپ کنید.

renamed successfully

باز کردن

با دستور زیر، فایل مذکور را باز می‌کنیم و وارد فایل می‌شویم.

```
open -name <file_name>
```

- اگر فایل مذکور موجود نبود، خطای زیر را چاپ کنید.

no such file

- اگر فایل مذکور، قابل باز کردن نبود (فایل زیپ یا فایل مخفی) خطای زیر را چاپ کنید.

unable to open file

- اگر خطایی رخ نداد، پیام زیر را چاپ کرده و وارد فایل می‌شویم.

entered file <file_name>

انتخاب کردن

با دستور زیر، دایرکتوری های نام برده شده که تعدادشان N است (N عددی بین 1 تا 9 است)، در حالت سلکت قرار می‌گیرند.

```
select -dir -N <number> -names:name1,name2,...,nameN
```

- اگر N عددی نامعتبر بود، خطای زیر را چاپ کنید.

invalid number

- اگر تعداد نام های داده شده با N متناقض بود، خطای زیر را چاپ کنید.

invalid number of arguments

- به ازای هر کدام از دایرکتوری ها که در دایرکتوری فعلی موجود نبود، خطای زیر را چاپ کنید. (مثلا اگر 3 دایرکتوری وجود نداشتند، در 3 خط خطا چاپ می‌شود.)

directory <directory_name> doesn't exist

با دستور زیر، فایل های نام برده شده که تعدادشان N است (N عددی بین 1 تا 9 است)، در حالت سلکت قرار می گیرند.

```
select -file -N <number> -names:name1,name2,...,nameN
```

- اگر N عددی نامعتبر بود، خطای زیر را چاپ کنید.

```
invalid number
```

- اگر تعداد نام های داده شده با N متناقض بود، خطای زیر را چاپ کنید.

```
invalid number of arguments
```

- به ازای هر کدام از فایل ها که در دایرکتوری فعلی موجود نبود، خطای زیر را چاپ کنید. (مثلا اگر 3 فایل وجود نداشتند، در 3 خط خطا چاپ می شود).

```
file <file_name> doesn't exist
```

- اگر خطایی رخ نداد، پیام زیر را چاپ کنید.

```
selected successfully
```

تمام دایرکتوری ها و فایل های سلکت شده، تنها تا 2 عملیات بعدی یا سلکت بعدی از همان جنس، معتبرند و پس از آن، دوباره باید در صورت نیاز آنها را سلکت کرد. (برای مثال برای زیپ کردن یک سری دایرکتوری و فایل باید در یک دستور، فایل ها و در یک دستور، دایرکتوری ها سلکت شوند و پس از آن بلافاصله عملیات زیپ کردن، اعمال شود).

همچنین محتویات دیلیت شده به وضوح از حالت سلکت خارج می شوند.

حذف

با دستور زیر، تمامی دایرکتوری های سلکت شده و تمام محتویات آنها، حذف می‌شود.

```
delete -dir
```

- اگر چیزی سلکت نشده بود، خطای زیر را چاپ کنید.

```
no directory selected
```

با دستور زیر، تمامی فایل های سلکت شده و تمام محتویات آنها، حذف می‌شود.

```
delete -files
```

- اگر چیزی سلکت نشده بود، خطای زیر را چاپ کنید.

```
no file selected
```

- اگر خطایی رخ نداد، پیام زیر را چاپ کنید.

```
deleted successfully
```

زیپ کردن

با دستور زیر، تمامی دایرکتوری ها و فایل هایی که حالت انتخاب شده دارند، درون فایلی با نام مشخص شده و پسوند zip. ریخته می‌شوند و فایل زیپ درون دایرکتوری فعلی قرار می‌گیرد.

```
zip -name <zip_name>
```

- اگر اسم فایل زیپ تکراری بود، خطای زیر را چاپ کنید.

```
file already exists
```

- اگر چیزی انتخاب نشده بود، خطای زیر را چاپ کنید.

nothing to zip

- اگر خطایی رخ نداد، پیام زیر را چاپ کنید.

zipped successfully

نکته: فایل زیپ، با دستور open باز نمی‌شود. همچنین با دستوری جز دستور zip نمی‌توان فایل زیپ ساخت.

همچنین فرمت نام فایل های زیپ مانند فرمت نام فایل های عادی است و در صورت مشاهده خطا در نام گذاری، خطای مربوطه را چاپ کنید.

برای دسترسی به محتویات فایل زیپ، باید آن را با دستور زیر، unzip کرد.

unzip -name <file_name>

- اگر نام مشخص شده وجود نداشت، خطای زیر را چاپ کنید.

no such file or directory

- اگر خطایی رخ نداد، پیام زیر را چاپ کنید.

unzipped successfully

نمایش

با دستور زیر، تمام دایرکتوری ها و فایل های درون دایرکتوری فعلی را به شکل زیر نمایش می‌دهیم.

show current content

```

dir:
1. name1
2. name2
...
file:
1. name1
2. name2
...
zip:
1. name1
2. name2
...

```

- اگر دایرکتوری فعلی خالی بود، پیام زیر را چاپ کنید.

directory empty

- هنگام نمایش، به این نکته دقت کنید که اگر مثلاً فایل a زودتر از فایل b ساخته شده بود، اگر فایل a زیپ شود و دوباره unzip شود، فایل a را باید بعد از فایل b نمایش دهید.

انتقال

با دستور زیر، تمامی موارد سلکت شده را به دایرکتوری نام برده شده جابجا می‌کنیم.

```
cut -dir <dir_name>
```

با دستور زیر، تمامی موارد سلکت شده را به دایرکتوری نام برده شده کپی می‌کنیم.

```
copy -dir <dir_name>
```

- اگر چیزی سلکت نشده بود خطای زیر را چاپ کنید.

nothing selected

- اگر دایرکتوری مورد نظر موجود نبود یا مخفی بود، خطای زیر را چاپ کنید.

invalid destination

- اگر دایرکتوری ای سلکت شده بود و دستور انتقال انجام شد، تمامی محتویات آن نیز انتقال می‌یابند.
- اگر مقصد عملیات، یکی از دایرکتوری های سلکت شده (مستقیم یا غیر مستقیم) بود، خطای زیر را چاپ کنید.

can't do action with a selected directory as destination

- اگر فایل یا دایرکتوری ای با نام مشابه وجود داشت، به انتهای نام آن، کاراکتر _ و سپس نام دایرکتوری مبدا را اضافه می‌کنیم. به طور مثال اگر فایل martin.txt از دایرکتوری me به دایرکتوری you منتقل شود و در آنجا نیز فایلی با همین نام داشتیم، در نهایت خواهیم داشت:

.../you/matin_me.txt

- اگر خطایی رخ نداد، پیام زیر را چاپ کنید. (copy برای عملیات کپی و cut برای عملیات cut)

(copy/cut) successfully

بازگشت

با دستور زیر، به منوی کاربری برمی‌گردیم و دیگر دستورات فاز 1 مورد قبول نخواهند بود.

go to user menu

- پس از این دستور، پیام زیر را چاپ کنید.

welcome to user menu

- نکته: اگر دستوری با دستورات گفته شده در این فاز، مطابقت نداشت، خطای زیر را چاپ کنید و دستور بعدی را دریافت کنید.

invalid command

فاز 2

بعد از دستور open وارد فایل می‌شویم. اگر فایل مذکور به فرمت های c، txt و java نبود، تنها قابلیت show details برای آن امکان پذیر است و در صورتی که دستور دیگری وارد شود، خطای زیر را چاپ کنید.

invalid command

جابجایی curser

در ابتدا، curser روی پوزیشن صفر اولین خط قرار دارد. با دستور زیر، می‌توان curser را به جهت مشخص شده و به تعداد مشخص شده جابجا کرد.

```
move_curser -direction <direction> -distance <distance>
```

جهت حرکت می‌تواند یکی از 4 جهت up و down و left و right باشد. همچنین عدد فاصله عددی بین 1 تا 10 است.

- اگر جهت چیزی جز این 4 جهت بود، خطای زیر را چاپ کنید.

invalid direction

- اگر عدد فاصله چیزی غیر از اعداد 1 تا 10 بود، خطای زیر را چاپ کنید.

invalid distance

- اگر مقادیر وارد شده باعث شد curser به جایی برود که خارج از محدوده فایل است، خطای زیر را چاپ کنید. همچنین در فایل خالی، curser در خط 1 و جایگاه 0 است و نمیتواند حرکت کند.

curser out of bounds

- در صورت رخ ندادن هیچگونه خطا، پیام زیر را چاپ کنید.

curser is in line <line_number> and position <position_number>

انتخاب متن

با دستور زیر، از مکان curser تا مکان مشخص شده را انتخاب میکنیم.

```
select -line <line_number> -pos <position>
```

با این دستور تمامی کاراکترهای بین curser تا مکان مشخص شده انتخاب میشوند. پس از اجرای دستور بعدی، تمامی موارد سلکت شده از حالت سلکت در می آیند.

- اگر مکان مشخص شده خارج از محدوده فایل بود، خطای زیر را چاپ کنید.

position out of bounds

- در صورت رخ ندادن هیچگونه خطا، پیام زیر را چاپ کنید.

selected successfully

حذف

با دستور زیر، تمامی محتوای انتخاب شده را از فایل حذف میکنیم.

delete

- اگر چیزی سلکت نشده بود، خطای زیر را چاپ کنید.

nothing to delete

- در صورت رخ ندادن هیچگونه خطا، پیام زیر را چاپ کنید.

deleted successfully

برش

با دستور زیر، تمامی محتوای انتخاب شده را کات می‌کنیم.

cut

- اگر چیزی سلکت نشده بود، خطای زیر را چاپ کنید.

nothing to cut

- در صورت رخ ندادن هیچگونه خطا، پیام زیر را چاپ کنید.

cut successfully

کپی

با دستور زیر، تمامی محتوای انتخاب شده را کپی می‌کنیم.

copy

- اگر چیزی سلکت نشده بود، خطای زیر را چاپ کنید.

nothing to copy

- در صورت رخ ندادن هیچگونه خطا، پیام زیر را چاپ کنید.

copied successfully

جایگذاری

با دستور زیر، موارد کات یا کپی شده را در محل curser پیست می‌کنیم.

paste

- اگر موردی برای پیست وجود نداشت، خطای زیر را چاپ کنید.

nothing to paste

- در صورت رخ ندادن هیچگونه خطا، پیام زیر را چاپ کنید.

pasted successfully

نوشتن

با دستور زیر، رشته وارد شده را در محل curser می‌نویسیم.

write -string "your string"

- اگر طول رشته بیش از 80 کاراکتر بود، خطای زیر را چاپ کنید.

string is too big!

- در تمامی دستورات بالا، اگر از متن قبل از curser، به میزان n کاراکتر حذف شود، curser به n کاراکتر قبل می‌رود و اگر n کاراکتر به بعد از آن اضافه شود، curser به n کاراکتر بعد می‌رود.
- در صورت رخ ندادن هیچگونه خطا، پیام زیر را چاپ کنید.

written successfully

نمایش جزییات

با دستور زیر، جزییات زیر از فایل به ترتیب و هر کدام در یک خط نمایش داده می‌شوند.

show details

۱. نام خام فایل (مثلا martin.txt)

۲. نام دایرکتوری ای که فایل در آن قرار دارد.

۳. نام کاربرانی که به فایل دسترسی دارند به ترتیب الفبای انگلیسی

۴. در صورتی که فرمت فایل، txt و java و c بود، محتوای فایل و در غیر اینصورت عبارت روبرو را چاپ کنید:

file content:

unable to show content

مثال:

```
file name: root/martin.txt
parent directory: root/matin
users with access:
1. aaalii
2. aalii
3. ali
4. li
5. martin
...
file content:
<file content> or the above message
```

میزان اشتراک

با دستور زیر، مشخص می‌شود که تغییرات اعمال شده، برای همه کاربرانی که به فایل دسترسی دارند اعمال شود یا صرفاً برای کاربری که فایل را تغییر داده اعمال شوند.

`apply changes for (me/all)`

- با انتخاب `me` فقط برای خود کاربر و با انتخاب `all`، تغییرات برای همه کاربران اعمال می‌شود.
- با توجه به آپشن انتخابی، پیام زیر را چاپ کنید.

`changes applied for (me/all)`

- اگر این دستور اجرا نشود، هیچکدام از تغییرات اعمال شده روی فایل انجام نمی‌گیرند و محتویات آن دست نخورده می‌ماند.

خروج

با دستور زیر، از فایل خارج می‌شویم و به دایرکتوری ای که فایل در آن واقع شده باز می‌گردیم.

`close`

پس از دستور بالا، پیام زیر را چاپ کنید.

`file closed. You are now in directory <parent_dir_name>`

توضیحات

- نام فایل ها و دایرکتوری ها، همه جا بجز مرحله ساخت، به صورت نام مسیر کامل از `root` آورده می‌شود. (در بخش تغییر نام، تضمین می‌شود که صرفاً نامی که فایل یا دایرکتوری با آن ساخته شده تغییر می‌کند و بقیه بخش های مسیر تغییری نمی‌کنند)

در زیر، یک نمونه از ورودی و خروجی مورد انتظار آورده شده است.

ورودی نمونه

```
create_user -username matin -password Mm9818650
create_user -username dorsa -password Mm9818650 -one_time
login -username dorsa -password Mm9818650
manage files
create -dir main
create -file main -format c
open -name root/main.c
write -string "#include<iostream>"
move_curser -direction left -distance 17
select -line 1 -pos 0
delete
show details
apply changes for me
close
cd -name root/main
cd ..
go to user menu
share -file root/main.c -target matin
view -shared -username matin -me
logout
login -username matin -password Mm9818650
view -shared -username dorsa
exit
```

خروجی نمونه

```
register successful
register successful
login successful
you are now in directory root
directory created successfully!
file created successfully!
entered file root/main.c
written successfully
invalid distance
```


selected successfully
deleted successfully
file name: root/main.c
parent directory: root
users with access:
1. dorsa
file content:
changes applied for me
file closed. You are now in directory root
entered directory root/main
entered directory root
welcome to user menu
shared successfully
shared to martin by me:
dir:
file:
1. root/main.c
zip:
logged out successfully
account removed!
login successful
no such user exists