

فروشگاه لباس

یک فروشگاه لباس برای نگهداری اطلاعات لباس‌های خود از یک برنامه‌ی جاوا استفاده می‌کند. این فروشگاه چهار نوع لباس (Clothing) دارد که برای هر یک، یک کلاس در نظر گرفته شده است:

- جوراب (Socks)
- پیراهن (Shirt)
- شلوار (Pants)
- ژاکت (Jacket)

برای هر لباس نیز سه ویژگی مشترک نگهداری می‌شود:

- نام: name
- فصل مناسب برای پوشیدن لباس: season (بهار، تابستان، پاییز و زمستان)
- قیمت پایه: basePrice

یک رفتار مهم و تغییرپذیر در این برنامه، محاسبه‌ی قیمت هر لباس با توجه به تخفیف‌های مختلف (زمستانه، شب یلدا و ...) است که هم‌اکنون با این منطق پیاده‌سازی شده است:

- **در تخفیف تابستانه:** لباس‌های بهاری: ۴۰٪ تخفیف، لباس‌های تابستانی: ۵۰٪ تخفیف و لباس‌های زمستانی: ۳۰٪ تخفیف.
- **در تخفیف زمستانه:** لباس‌های پاییزی: ۴۰٪ تخفیف، لباس‌های زمستانی: ۵۰٪ تخفیف و لباس‌های تابستانی: ۲۵٪ تخفیف. به علاوه، تمامی ژاکت‌ها از ۱۰٪

تخفیف دیگر نیز برخوردار هستند. یعنی یک ژاکت پاییزی ابتدا ۴۰٪ از قیمتش کسر می‌شود چون لباس پاییزیست و سپس مجدداً ۱۰٪ دیگر نیز از قیمتش کاسته می‌شود.

- **در تخفیف شب یلدا:** تمامی ژاکت‌ها: ۱۰٪ تخفیف، تمامی جوراب‌ها: ۲۰٪ تخفیف و بقیه‌ی لباس‌ها: ۲۵٪ تخفیف.

برنامه‌نویس، در نسخه‌ی قدیمی این برنامه در کلاس Clothing یک متد به نام `getPrice` با ورودی نوع تخفیف (`discountType`) قرار داده بود که با توجه به نوع تخفیف، نوع لباس و ویژگی‌های لباس، قیمت نهایی را محاسبه می‌کرد:

```

1 public long getPrice(DiscountType discountType) {
2     if (discountType == DiscountType.YALDA) {
3         // return with YALDA discount
4     } else if (discountType == DiscountType.SUMMER) {
5         // return with SUMMER discount
6     } else if (discountType == DiscountType.WINTER) {
7         // return with WINTER discount
8     }
9     return price;
10 }
```

حال پس از چند ماه مسئول فروشگاه از او می‌خواهد چند نوع تخفیف جدید برای مناسبت‌های مختلف اضافه کرده و رفتار چند نوع تخفیف موجود را تغییر دهد. برنامه‌نویس متوجه مسأله‌ای می‌شود؛ با اضافه کردن این تغییرات به متد `getPrice` و تغییرات احتمالی آینده، این متد رفته‌رفته بزرگ‌تر و ناخوانا‌تر می‌شود. او با مطالعه‌ی سرفصل‌های چندریختی (Polymorphism) و واسط (Interface) از سایت جاواکاپ به ایده‌ی جدیدی می‌رسد. او برای شروع، متد `getPrice` خود را حذف می‌کند؛ ولی او این مباحث را به‌تازگی مطالعه کرده‌است، پس به او کمک کنید تا ایده‌ی خود را پیاده‌سازی کند.

ابتدا بسته‌ی [source](#) را دانلود کنید.

او می‌خواهد وظیفه‌ی محاسبه‌ی هر تخفیف را به کلاس‌های مجزایی بسپارد. برای این کار ابتدا یک واسط به نام DiscountStrategy ایجاد کنید:

```
1 public interface DiscountStrategy {  
2     long priceByDiscount(Clothing clothing);  
3 }
```

سپس به ازای هر نوع تخفیف، یک کلاس ایجاد کنید که این واسط را پیاده‌سازی می‌کند. در متد priceByDiscount هر یک از این کلاس‌ها باید رفتار نوع تخفیف مربوطه را پیاده‌سازی کنید:

- برای تخفیف تابستانه: کلاس SummerDiscountStrategy
- برای تخفیف زمستانه: کلاس WinterDiscountStrategy
- برای تخفیف شب یلدا: کلاس YaldaDiscountStrategy

در کلاس Clothing متد زیر را برای دریافت و ثبت **نوع تخفیف فعلی** پیاده‌سازی کنید:

```
1 public void setDiscountStrategy(DiscountStrategy discountStrategy) {  
2     // TODO set discountStrategy  
3 }
```

حال در متد getPrice جدید (که دیگر پارامتر ورودی ندارد) با استفاده از discountStrategy مشخص‌شده، قیمت با تخفیف را محاسبه کرده و برگردانید:

```
1 public long getPrice() {  
2     // TODO get price by considering discountStrategy  
3 }
```

اگر هیچ `discountStrategy` تعیین نشده بود، باید قیمت پایه‌ی لباس برگردانده شود. نکته: این متد نباید از نحوه‌ی محاسبه‌ی تخفیف‌های مختلف خبر داشته باشد. به عبارت دیگر، تمام منطق محاسبه‌ی هر نوع تخفیف باید در کلاس مربوط به خود نوشته شود.

مثال

```
1 Jacket linenCoat = new Jacket("Linen Coat", Season.FALL, 100000);  
2 System.out.println("Linen Coat: " + linenCoat.getPrice());  
3  
4 linenCoat.setDiscountStrategy(new YaldaDiscountStrategy());  
5 System.out.println("Linen Coat with Yalda discount: " + linenCoat.getPrice());  
6  
7 linenCoat.setDiscountStrategy(new WinterDiscountStrategy());  
8 System.out.println("Linen Coat with Winter discount: " + linenCoat.getPrice());  
9  
10 Clothing woolenSocks = new Pants("Woolen Socks", Season.WINTER, 10000);  
11 System.out.println("Woolen Socks: " + woolenSocks.getPrice());  
12  
13 woolenSocks.setDiscountStrategy(new SummerDiscountStrategy());  
14 System.out.println("Woolen Socks with Summer discount: " + woolenSocks.getPrice());  
15  
16  
17
```

```
woolenSocks.setDiscountStrategy(new WinterDiscountStrategy());  
System.out.println("Woolen Socks with Winter discount: " + woolenSocks.getPrice());
```

خروجی:

```
Linen Coat: 100000  
Linen Coat with Yalda discount: 90000  
Linen Coat with Winter discount: 54000  
Woolen Socks: 10000  
Woolen Socks with Summer discount: 7000  
Woolen Socks with Winter discount: 5000
```

آنچه باید آپلود کنید، یک فایل zip است. به صورتی که وقتی فایل زیپ را باز می‌کنیم، دقیقا پنج فایل با نام‌های زیر موجود باشد:

فایل Clothing.java

فایل DiscountStrategy.java

فایل SummerDiscountStrategy.java

فایل WinterDiscountStrategy.java

فایل YaldaDiscountStrategy.java

کلمه شمار

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

می‌خواهیم تعداد تکرار هر کلمه را در یک جمله پیدا کنیم.

با دریافت جمله‌ی موردنظر در ورودی، تعداد تکرار هر کلمه را در آن شمرده و براساس **ترتیب حروف الفبا** بین کلمات، تعداد تکرار هر کدام را چاپ کنید.

ورودی

در یک خط، یک جمله شامل حروف و اعداد انگلیسی داده می‌شود که کلمات آن با یک فاصله از هم جدا شده‌اند.

خروجی

تعداد کلمات جمله‌ی دریافت شده در ورودی را مطابق فرمت مثال‌ها چاپ کنید.

مثال

ورودی نمونه ۱

```
2ps0ZrCCzNhpn 2ps0ZrCCzNhpn 2ps0ZrCCzNhpn 2ps0ZrCCzNhpn 2ps0ZrCCzNhpn 2ps0ZrCCzNhpn 2ps0ZrCCzNhpn
2ps0ZrCCzNhpn 2ps0ZrCCzNhpn
```

خروجی نمونه ۱

```
{'2ps0ZrCCzNhpn' = 9}
```

ورودی نمونه ۲

```
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
```

خروجی نمونه ۲

```
{'1' = 2 , '2' = 2 , '3' = 2 , '4' = 2 , '5' = 2 , '6' = 2 , '7' = 2 , '8' = 2 , '9' = 1}
```

ورودی نمونه ۳

```
Salam salam sad ta salam khobi salam merC to khobi
```

خروجی نمونه ۳

```
{'Salam' = 1 , 'khobi' = 2 , 'merC' = 1 , 'sad' = 1 , 'salam' = 3 , 'ta' = 1 , 'to' = 1}
```


تاریخ

قرار است برای نگهداری تاریخ در یک برنامه‌ی بزرگ، یک کتابخانه برای مدیریت تاریخ‌های شمسی و میلادی بنویسید.

ابتدا فایل را از [source](#) دانلود کنید و به آن را به دقت نگاه کنید.

شما باید دو کلاس زیر را با توجه به نکات بیان‌شده، پیاده‌سازی کنید.

- تاریخ شمسی (ShamsiCustomCalendar)
- تاریخ میلادی (GregorianCustomCalendar)

هر کدام از تاریخ‌ها، تعدادی متد پایه دارند که در واسط CustomCalendar قابل مشاهده می‌باشند. برای آشنایی با عملکرد متدها، به نکات زیر توجه کنید:

```
1 | int getDay();  
2 | void setDay(int day);
```

برای تنظیم روز یک تاریخ، از این متدها استفاده می‌شود. اگر day با توجه به مقدار آن و ماه تاریخ، در محدوده‌ی معتبر نباشد، خطای IllegalArgumentException پرتاب می‌شود.

```
1 | int getMonth();  
2 |
```

```
void setMonth(int month);
```

برای تنظیم ماه یک تاریخ، از این متدها استفاده می‌شود. اگر month با توجه به مقدار آن و روز تاریخ، در محدوده‌ی معتبر نباشد، خطای IllegalArgumentException پرتاب می‌شود.

```
1 | int getYear();  
2 | void setYear(int year);
```

برای تنظیم سال یک تاریخ، از این متدها استفاده می‌شود. اگر year در محدوده‌ی معتبر نباشد، خطای IllegalArgumentException پرتاب می‌شود.

```
1 | String getMonthName();
```

ماه تاریخ را در فرمت رشته بر می‌گرداند.

```
1 | CustomCalendar increaseOneDay();  
2 | CustomCalendar decreaseOneDay();
```

این دو متد تاریخ را یک روز جلو یا عقب می‌برند و فراخوانی آن‌ها روی یک شی، باعث تغییر آن شی می‌شود. (حالت خاص 1/1/1 را برای کاهش روز در نظر بگیرید. 😊)

1 | `void setCalendarRepresentation(CalendarRepresentation representation);`

این متد نحوه‌ی نمایش و چاپ تاریخ را با توجه به شی CalendarRepresentation دریافت‌شده مشخص می‌کند. اگر به هنگام چاپ تاریخ، هیچ نوع نمایشی برای آن مشخص نشده بود، باید خطای IllegalStateException پرتاب شود.

1 | `public int compareTo(CustomCalendar object)`

شی تاریخ کنونی را با پارامتر دریافت شده مقایسه می‌کند، در صورت بزرگتر بودن شی کنونی مقدار مثبت، در صورت کوچکتر بودن آن مقدار منفی و در صورت برابری دو تاریخ صفر برمی‌گرداند. دقت کنید که اگر دو تاریخی که مقایسه می‌شوند از یک جنس نباشند یا ورودی نامعتبر باشد، خطای IllegalArgumentException پرتاب می‌شود.

- هر شی تاریخ را باید بتوان از روی یک رشته‌ی اولیه، ساخت. در صورت نامعتبر بودن آن رشته، خطای IllegalCalendarFormatException پرتاب می‌شود.
- برای نمایش تاریخ دو نوع CalendarRepresentation را به شکل زیر پیاده‌سازی کنید:

○ نمایش DateRepresentation: در این نمایش تاریخ در فرمت dd/mm/yyyy چاپ می‌شود. در این نمایش **روز و ماه** باید حتما **دو رقمی** باشند اما تعداد ارقام سال در آن **مهم نیست**

○ نمایش FullRepresentation: در این نمایش تاریخ در فرمت dd MM yyyy چاپ می‌شود. در این نمایش تعداد ارقام روز و سال نباید بیشتر از **تعداد لازم** باشد و ماه باید به شکل رشته نمایش داده شود.

- ماه‌های میلادی و شمسی را به صورت زیر در نظر بگیرید:

نام ماه	طول ماه	نام ماه	طول ماه
January	۳۱	Farvardin	۳۱
February	۲۸	Ordibehesht	۳۱
March	۳۱	Khordad	۳۱
April	۳۰	Tir	۳۱
May	۳۱	Mordad	۳۱
June	۳۰	Shahrivar	۳۱
July	۳۱	Mehr	۳۰
August	۳۱	Aban	۳۰
September	۳۰	Azar	۳۰
October	۳۱	Dey	۳۰
November	۳۰	Bahman	۳۰

نام ماه	طول ماه	نام ماه	طول ماه
December	۳۱	Esfand	۳۰

- در این برنامه بررسی سال کبیسه لازم نیست.

مثال

```
1 CustomCalendar shamsiCalendar = new ShamsiCustomCalendar("1/1/1397");
2 shamsiCalendar.setCalendarRepresentation(new DateRepresentation());
3 System.out.println(shamsiCalendar); // 01/01/1397
4 shamsiCalendar.setCalendarRepresentation(new FullRepresentation());
5 System.out.println(shamsiCalendar); // 1 Farvardin 1397
6 shamsiCalendar.increaseOneDay();
7 System.out.println(shamsiCalendar); // 2 Farvardin 1397
8 shamsiCalendar.decreaseOneDay().decreaseOneDay();
9 System.out.println(shamsiCalendar); // 30 Esfand 1396
10 shamsiCalendar.increaseOneDay();
11 System.out.println(shamsiCalendar); // 1 Farvardin 1397
12 try {
13     shamsiCalendar.setMonth(13);
14     System.out.println("Unable to catch IllegalArgumentException");
15 } catch (IllegalArgumentException e) {
16     System.out.println("Catch IllegalArgumentException");
17 }
```

```
}
shamsiCalendar.setDay(31);
try {
    shamsiCalendar.setMonth(7);
    System.out.println("Unable to catch IllegalArgumentException");
} catch (IllegalArgumentException e) {
    System.out.println("Catch IllegalArgumentException");
}
shamsiCalendar.setCalendarRepresentation(new DateRepresentation());
System.out.println(shamsiCalendar); // 31/01/1397
shamsiCalendar.setDay(30);
shamsiCalendar.setMonth(12);
shamsiCalendar.increaseOneDay().increaseOneDay().increaseOneDay().decreaseOneDay();
System.out.println(shamsiCalendar); // 02/01/1398

try {
    CustomCalendar gregorianCalendar = new GregorianCustomCalendar("salam");
    System.out.println("Unable to catch IllegalCalendarFormatException");
} catch (IllegalCalendarFormatException e) {
    System.out.println("Catch IllegalCalendarFormatException");
}
CustomCalendar gregorianCalendar = new GregorianCustomCalendar("01/1/1980");
try {
    System.out.println(gregorianCalendar); // representation must be set before print
    System.out.println("Unable to catch IllegalStateException");
} catch (IllegalStateException e) {
    System.out.println("Catch IllegalStateException");
}
```

```
}
gregorianCalendar.setCalendarRepresentation(new DateRepresentation());
System.out.println(gregorianCalendar); // 01/01/1980
gregorianCalendar.setYear(2020);
gregorianCalendar.setMonth(12);
while (gregorianCalendar.getMonth() != 1)
    gregorianCalendar.increaseOneDay();
System.out.println(gregorianCalendar); // 01/01/2021

System.out.println(gregorianCalendar.compareTo(new GregorianCalendar("02/01/2021")) < 0); // tru
System.out.println(gregorianCalendar.compareTo(new GregorianCalendar("01/02/2021")) < 0); // tru
System.out.println(gregorianCalendar.compareTo(new GregorianCalendar("15/12/2020")) > 0); // tru
System.out.println(gregorianCalendar.compareTo(gregorianCalendar)); // 0
```

آن چه که باید آپلود کنید

یک فایل zip است که وقتی آن را باز می‌کنیم، در آن دو پوشه‌ی calendar و representation باشد. در پوشه‌ی calendar فقط دو فایل FullRepresentation و DateRepresentation و در پوشه‌ی representation فقط دو فایل GregorianCalendar و ShamsiCustomCalendar باشد.

لیست گپ

- محدودیت زمان: ۱۰ ثانیه

- محدودیت حافظه: ۲۵۶ مگابایت

می‌خواهیم سرور یک برنامه‌ی گپ را بنویسیم که مدیریت کاربرانی که به برنامه می‌پیوندند و دستوراتی که به آن ارسال می‌شود را به عهده دارد.

در پیاده‌سازی خود به نکات زیر توجه کنید:

- هر کاربر دارای نام و نام خانوادگی، **صفر تا چند** شماره‌ی متمایز از بقیه‌ی کاربران و لیستی از مخاطبان در برنامه‌ی گپ می‌باشد. در سرور، یک لیست از کاربرانی که به برنامه‌ی گپ پیوسته‌اند، نگه داشته می‌شود.
- دو کاربر نمی‌توانند هم‌زمان نام و نام خانوادگی یکسان داشته باشند (نام و نام خانوادگی به حروف کوچک و بزرگ حساس **نیستند**)، همچنین هیچ دو کاربری دارای شماره‌ی مشترک نیستند. بنابراین **نباید** امکان اضافه کردن کاربر تکراری وجود داشته باشد.
- مرتب‌سازی و نمایش لیست تمام کاربران برنامه و یا مخاطبان یک کاربر به دو صورت انجام می‌شود:
 - بر اساس نام : اگر نام دو کاربر یکسان بود، باید براساس نام‌خانوادگی مرتب شوند.
 - بر اساس نام‌خانوادگی: اگر نام خانوادگی دو کاربر یکسان بود، باید براساس نام مرتب شوند.
- مقایسه در مرتب‌سازی، حساس به حروف کوچک و بزرگ **نیست**.
- هر کاربر می‌تواند مخاطبش را block کند. تا زمانی که آن مخاطب unblock نشده باشد، **نباید** در نمایش لیست مخاطبان کاربر نشان داده شود.
- هر کاربر می‌تواند مخاطبان خاص هم داشته باشد!! در نمایش لیست مخاطبان کاربر، ابتدا این دسته مخاطبان به صورت مرتب شده می‌آیند و بعد سایر مخاطبان

به صورت مرتب شده بعد از مخاطبان خاص نمایش داده می‌شوند.

- تعداد کل کاربران شبکه‌ی اجتماعی از ۲۰۰۰ تا بیش‌تر نمی‌شود.
- تعداد شماره‌های یک فرد حداکثر ۱۰ مورد و تعداد contact های یک فرد حداکثر ۱۰۰۰ فرد است

درخواست‌های سرور شامل موارد زیر می‌باشد:

```
join [first-name] [last-name] [phone numbers]
```

با این درخواست، کاربری با نام و نام‌خانوادگی دریافت‌شده به سرور اضافه می‌شود. قسمت شماره‌ها می‌تواند خالی باشد و برای اضافه‌شدن کاربر تکراری، باید پیام joinError چاپ شود.

```
search [first-name] [last-name]
search [phone number]
```

جستجو برای یک کاربر را می‌توان به دو شکل بالا انجام داد. در صورت یافتن شخص، باید نام، نام‌خانوادگی و شماره‌های او با توجه به مثال‌ها چاپ شود و در غیر این صورت باید پیام searchError چاپ شود.

```
addContact [user]: [contact]
```

با این درخواست، کاربر contact به لیست مخاطبین user اضافه می‌شود. user و contact می‌توانند با نام و نام‌خانوادگی یا یکی از شماره‌هایشان داده شوند. در

صورت عدم وجود user یا contact، یکسان بودن user و contact یا قرار داشتن contact در لیست مخاطبان user، باید پیام addContactError چاپ شود.

```
remove [first-name] [last-name]
remove [phone number]
```

این درخواست، کاربر مشخص شده را از سرور و لیست مخاطبان سایر کاربران حذف می‌کند. در صورتی که کاربر وجود نداشت، باید پیام removeError چاپ شود.

```
listUsers [first-name | last-name]
```

با این درخواست، لیست **مرتب‌شده‌ی** تمام کاربران برنامه براساس نوع مقایسه‌ی مشخص شده چاپ می‌شود.

```
block [user]: [contact]
unblock [user]: [contact]
```

با این درخواست‌ها، block یا unblock شدن کاربر contact از طرف user مدیریت می‌شود. user و contact می‌توانند با نام و نام خانوادگی یا یکی از شماره‌هایشان

داده شوند. در صورت عدم وجود user یا contact یا نبودن contact در لیست مخاطبان user، باید پیام مناسب blockError یا unblockError چاپ شود.

اجرای دستور block برای یک **مخاطب خاص**، او را از وضعیت خاص بیرون آورده و بعد بلاک می‌کند.

اجرای دستور unblock برای یک مخاطب، او را به **مخاطب عادی** تبدیل می‌کند.

```
specific [user]: [contact]
```

```
unspecific [user]: [contact]
```

با این درخواست‌ها، specific یا unspecific شدن کاربر contact از طرف user مدیریت می‌شود. user و contact می‌توانند با نام و نام خانوادگی یا یکی از شماره‌هایشان داده شوند. در صورت عدم وجود user یا contact یا نبودن contact در لیست مخاطبان user، باید پیام مناسب specificError یا unspecificError چاپ شود.

اجرای دستور specific برای یک مخاطب بلاک شده، او را از وضعیت بلاک بیرون آورده و بعد خاص می‌کند. اجرای دستور unspecific برای یک مخاطب، او را به مخاطب عادی تبدیل می‌کند.

```
listContacts [first-name | last-name] [user]
```

این درخواست، لیست مرتب‌شده‌ی مخاطبان user را بر اساس نوع مقایسه‌ی مشخص شده و نکته‌های قبلی چاپ می‌کند. user می‌تواند با نام و نام خانوادگی یا یکی از شماره‌هایش داده شود و در صورت عدم وجود او، باید پیام listContactsError چاپ شود.

نکته: شماره‌ها دارای رقم صفر هستند و بین شماره‌های 0919 و 00919 و 919 تفاوت وجود دارد!

ورودی

تا زمانی که از طرف برنامه به سرور درخواست می‌آید، درخواست‌ها را مدیریت کرده و پیام‌های مناسب چاپ کنید.

خروجی

خروجی مطابق توضیحات داده شده است.

مثال

ورودی نمونه ۱

```
join Ali alavi 09123456789 03538234753
search ali alavi
search 09123456789
join ali Alavi 123
join taghi taghavi 09123456789 95135742
join amin bor 09134679852 684555 654599746
join mamad mamadi 987652
join reza nasab 7419638520
join ali mohammady
addContact ali alavi: mamad mamadi
addContact ali alavi: 09134679852
addContact ali alavi: 3
listContacts first-name ali alavi
remove 09134679852
listContacts first-name ali alavi
listUsers last-name
block ali alavi: mamad mamadi
listContacts first-name ali alavi
specific ali alavi: mamad mamadi
```

```
listContacts first-name ali alavi
specific ali alavi: 3
search new username
join new username
search new username
listUsers last-name
remove 1 1
```

خروجی نمونه ۱

```
Ali alavi [09123456789, 03538234753]
Ali alavi [09123456789, 03538234753]
joinError
joinError
addContactError
amin bor [09134679852, 684555, 654599746]
mamad mamadi [987652]
mamad mamadi [987652]
Ali alavi [09123456789, 03538234753]
mamad mamadi [987652]
ali mohammady []
reza nasab [7419638520]
mamad mamadi [987652]
specificError
searchError
new username []
```

```
Ali alavi [09123456789, 03538234753]
mamad mamadi [987652]
ali mohammady []
reza nasab [7419638520]
new username []
removeError
```

ورودی نمونه ۲

```
search 1150
join Bryson Ryan
join Bryson Ryan
join Hunter Julian 1150
addContact 1150: 1150
join Bryson Ryan
unspecific 3846276423: Theodore Charles
addContact Theodore Charles: Theodore Charles
search Hunter Julian
addContact 6211446: 609997780
block Dominic Dylan: Dominic Dylan
block Carson Jordan: 3846276423
remove 5438670
addContact Theodore Charles: 609997780
listUsers first-name
remove 1150
listContacts last-name 6211446
```

```
listContacts first-name 7
unlock Jack Sawyer: 320472
listContacts last-name Alexander Kayden
listContacts last-name 410
addContact 351: Jack Sawyer
search Samuel Adrian
join William Gavin
join Henry Jacob
join Santiago James 422775347 0
listContacts last-name 3846276423
listContacts last-name Jack Sawyer
listUsers last-name
listUsers first-name
join Carson Jordan
join Levi Isaiah 410 3846276423
addContact 0006: Lucas Aaron
join Isaiah Parker
addContact Jack Sawyer: 422775347
```

خروجی نمونه ۲

```
searchError
joinError
addContactError
joinError
unspecificError
```

```
addContactError
Hunter Julian [1150]
addContactError
blockError
blockError
removeError
addContactError
Bryson Ryan []
Hunter Julian [1150]
listContactsError
listContactsError
unblockError
listContactsError
listContactsError
addContactError
searchError
listContactsError
listContactsError
William Gavin []
Henry Jacob []
Santiago James [422775347, 0]
Bryson Ryan []
Bryson Ryan []
Henry Jacob []
Santiago James [422775347, 0]
William Gavin []
addContactError
```


addContactError

ورودی نمونه ۳

```
search 1150
join Bryson Ryan
join Bryson Ryan
join Hunter Julian 1150
addContact 1150: 1150
join Bryson Ryan
unspecific 3846276423: Theodore Charles
addContact Theodore Charles: Theodore Charles
search Hunter Julian
addContact 6211446: 609997780
block Dominic Dylan: Dominic Dylan
block Carson Jordan: 3846276423
remove 5438670
addContact Theodore Charles: 609997780
listUsers first-name
remove 1150
listContacts last-name 6211446
listContacts first-name 7
unblock Jack Sawyer: 320472
listContacts last-name Alexander Kayden
listContacts last-name 410
addContact 351: Jack Sawyer
```

```
search Samuel Adrian
join William Gavin
join Henry Jacob
join Santiago James 422775347 0
listContacts last-name 3846276423
listContacts last-name Jack Sawyer
listUsers last-name
listUsers first-name
join Carson Jordan
join Levi Isaiah 410 3846276423
addContact 0006: Lucas Aaron
join Isaiah Parker
addContact Jack Sawyer: 422775347
```

خروجی نمونه ۳

```
searchError
joinError
addContactError
joinError
unspecificError
addContactError
Hunter Julian [1150]
addContactError
blockError
blockError
```

```
removeError
addContactError
Bryson Ryan []
Hunter Julian [1150]
listContactsError
listContactsError
unlockError
listContactsError
listContactsError
addContactError
searchError
listContactsError
listContactsError
William Gavin []
Henry Jacob []
Santiago James [422775347, 0]
Bryson Ryan []
Bryson Ryan []
Henry Jacob []
Santiago James [422775347, 0]
William Gavin []
addContactError
addContactError
```

ورودی نمونه ۴

```
join Luke Charles
```

```
unspecific Jameson Mateo: Nolan Dominic
addContact Jack James: 621342431
unspecific 1789066805: Parker Michael
join Santiago Nolan 10927
search 345519848
join Ian Joshua
addContact 3410910630: Nicholas Owen
join Jack Logan 799
listUsers first-name
addContact Nolan William: Lincoln Jonathan
join Dominic Ezra
join Liam Wyatt 1370313 31412267
listContacts last-name Lincoln Noah
listContacts first-name 427746198
listContacts last-name Ayden Bryson
listContacts last-name 799
listUsers last-name
unspecific Mason Levi: Ayden Dominic
join Joshua Aiden
join Evan Jonathan
join Luke Christian
listContacts first-name Mateo Austin
addContact 6581: Maverick Mason
listContacts last-name 37179
specific 10927: Jacob Carter
unblock 621342431: 50
remove Robert Theodore
```

```
block 98492199: 602301
listContacts last-name 2
block 6: 4363955
addContact 5732732856: 16379022
join Aaron Jordan
unspecific 013745: 73
remove 2
unblock Gavin Gavin: Carson Jaxon
remove 1155
join Ezekiel Jeremiah 745922226 024084068
specific 37179: Cooper Samuel
unblock 6: 235
block Austin David: Nolan Christopher
join David Jameson
join Jack Asher 4
join Nolan Christopher
unblock 02265767: 0518932458
listUsers last-name
addContact 15592: 464468
join Jace Jaxon 180110353
addContact 37293: Gavin Gavin
listContacts first-name John Christian
join Cameron Josiah
search Mason Logan
addContact Austin David: Asher Kayden
specific Jonathan Andrew: Adam Jacob
join Ian Joshua
```

```
join Christian Caleb
unspecific Aaron Jacob: 113817
join Elijah Julian
join Jose John
join Ian Landon
addContact 080060224: 0993977
listContacts first-name Jackson Landon
listContacts last-name Nolan Dominic
listUsers first-name
block 00436: 89813331
search Jacob David
join Anthony Isaiah
specific 2651: William Landon
listContacts first-name Xavier Owen
listUsers first-name
```

خروجی نمونه ۴

```
unspecificError
addContactError
unspecificError
searchError
addContactError
Ian Joshua []
Jack Logan [799]
Luke Charles []
```

Santiago Nolan [10927]
addContactError
listContactsError
listContactsError
listContactsError
Luke Charles []
Dominic Ezra []
Ian Joshua []
Jack Logan [799]
Santiago Nolan [10927]
Liam Wyatt [1370313, 31412267]
unspecificError
listContactsError
addContactError
listContactsError
specificError
unblockError
removeError
blockError
listContactsError
blockError
addContactError
unspecificError
removeError
unblockError
removeError
specificError

```
unblockError
blockError
unblockError
Joshua Aiden []
Jack Asher [4]
Luke Charles []
Luke Christian []
Nolan Christopher []
Dominic Ezra []
David Jameson []
Ezekiel Jeremiah [745922226, 024084068]
Evan Jonathan []
Aaron Jordan []
Ian Joshua []
Jack Logan [799]
Santiago Nolan [10927]
Liam Wyatt [1370313, 31412267]
addContactError
addContactError
listContactsError
searchError
addContactError
specificError
joinError
unspecificError
addContactError
listContactsError
```



```
listContactsError
Aaron Jordan []
Cameron Josiah []
Christian Caleb []
David Jameson []
Dominic Ezra []
Elijah Julian []
Evan Jonathan []
Ezekiel Jeremiah [745922226, 024084068]
Ian Joshua []
Ian Landon []
Jace Jaxon [180110353]
Jack Asher [4]
Jack Logan [799]
Jose John []
Joshua Aiden []
Liam Wyatt [1370313, 31412267]
Luke Charles []
Luke Christian []
Nolan Christopher []
Santiago Nolan [10927]
blockError
searchError
specificError
listContactsError
Aaron Jordan []
Anthony Isaiah []
```

Cameron Josiah []
Christian Caleb []
David Jameson []
Dominic Ezra []
Elijah Julian []
Evan Jonathan []
Ezekiel Jeremiah [745922226, 024084068]
Ian Joshua []
Ian Landon []
Jace Jaxon [180110353]
Jack Asher [4]
Jack Logan [799]
Jose John []
Joshua Aiden []
Liam Wyatt [1370313, 31412267]
Luke Charles []
Luke Christian []
Nolan Christopher []
Santiago Nolan [10927]