دانشکده مهندسی کامپیوتر

برنامه سازی پیشرفته

نیم‌سال اول 93-94

پیمان دودانگه

تمرین سوم

زمان تحویل:12 دی

1. موعد ارسال تمرین ساعت 23:59 روز جمعه مورخ 1393/10/12 است.
2. تمرین باید به صورت حضوری تحویل داده شود و استفاده از کامنت، و ساختار کد نمره خواهد داشت.
3. در صورت داشتن هرگونه مشکل، سوال خود را با تگ مناسب در Piazza مطرح کنید.
4. با هر گونه تقلب برخورد جدی خواهد شد.
5. برای تحویل تمرین ، آن را بر روی سامانه ی داوری آپلود نمایید. بعد از آپلود ، شما از سامانه داوری نمره ای دریافت نمی کنید . آپلود تنها به منزله ی ارسال تمرین شما می باشد.

CECollection

تیم تخصّصی بررسی کننده‌ی کارایی‌های ساختی ابزارهای تسهیل‌کننده‌ی زبان‌های برنامه نویسی ساخت‌یافته‌ی دانشکده‌ی مهندسی کامپیوتر دانشگاه صنعتی شریف پس از پژوهش‌های فراوان در حوزه‌ی ابزارهای درونی جاوا و کاربردهای آن به این نتیجه رسید که کلاسهایی که به عنوان Collection در جاوا از آن‌ها یاد می‌شود آن‌چنان که شایسته نام جاواست، پیاده‌سازی نشده‌اند و نیاز به بازبینی فراوان دارند!

بس از بررسی‌های کارشناسی بیشتر در کمیته‌های تخصصی تر، این تیم (که دیگر از ذکر نام آن خودداری می‌کنیم!) به این نتیجه رسید که مشکلات این کلاس‌ها بنیادی تر از آن است که بتوان با اصلاحاتی جزئی آن‌ها را رفع رجوع کرد. بنابر این تصمیم بر این گرفته شد که Collection و زیر مجموعه های آن توسط خود بچه‌های دانشکده به خصوص دانشجویان درس برنامه‌نویسی پیشرفته با عنوان CECollection بازنویسی شود تا این ایرادات اصلاح شده و جامعه جهانی بتواند با آرامش غیر قابل وصفی از این ابزارهای بی‌نقص استفاده کند و دانشجویان این درس هم تمرینی روی مفاهیم آن انجام داده باشند!

از آن‌جایی که این تیم به هیچ یک از دیگر ابزارهای درونی نوشته شده‌ی جاوا اعتماد ندارد، ابزارهای جدید صرفا باید از امکانات مقدماتی جاوا مثل آرایه و تابع و ... استفاده کنند و از هیچ یک از کلاس‌های آماده جاوا در این برنامه‌ها نباید استفاده کرد.

توجه: در صورت استفاده از هر کلاس از پیش نوشته شده هیچ نمره‌ای به این سوال تعلق نمی‌گیرد!

شما باید در این برنامه همه classها و interfaceها و توابع خواسته شده را پیاده‌سازی کنید.

```
public interface CEComparable<T>;
```

همه کلاس‌هایی که اعضای آن با اعضای کلاس جنریکی مانند T قابل مقایسه اند، این interface را implement می‌کنند. توجه کنید که در مثالهای ما T همان کلاسی است که CEComparable را implement کرده است. به طور مثال

```
class Student implement CECoparable<Student>
```

یعنی Student با خودش قابل مقایسه است.

*صرفا جهت اطلاع: T از آن جهت به CEComparable داده شده که در تابع ComparetTo که در خط بعد تعریف شده بتواند به عنوان تایپ قرار بگیرد!

این interface تنها تابع زیر را داراست:

```
public int CompareTo(T a);
```

خروجی این تابع، در صورتی که شیئی که این تابع روی آن صدا زده شده از a کوچکتر باشد برابر 1-، در صورتی که هر دو مساوی باشند 0 و در صورتی که a از شیئی که این تابع روی آن صدا زده شده کوچکتر باشد برابر 1 خواهد شد.

در مثال Student که در بالا زده شد، تابع CompareTo به شکل زیر در می‌آید:

```
public int CompareTo(Student a);
```

که اگر s و r دو نمونه از کلاس student باشند میتوان تابع را این گونه صدا کرد:

```
s.CompareTo(r);
```

در صورتی که از لحاظ معیار در نظر گرفته‌شده در تابع compareTo، s کوچکتر از r باشد حاصل 1- اگر s و r مساوی باشند حاصل 0 و اگر s بزرگتر از r باشد حاصل 1 خواهد شد.

```
public interface CECollection<E>
```

همه کلاس‌هایی که در ادامه می‌آیند به شکل مستقیم یا غیر مستقیم باید این interface را implement کنند. در واقع همه کلاس-هایی که مجموعه‌ای از instance های یک کلاس جنریک (E) را قرار است نگه‌داری کنند، این interface را implement می‌کنند. (همان طور که همه کلاس‌هایی مثل Arraylist یا Set که مجموعه‌ای از نمونه‌های یک کلاس را نگه داری می‌کنند کلاس Collection جاوا را implement می‌کنند.)

این interface توابع زیر را داراست:

```
public void add(E a);
```

این تابع a را به CECollection مربوطه اضافه می‌کند.

```
public void add(E... a);
```

این تابع تعداد نامعلومی نمونه از کلاس E را به CECollection مربوطه اضافه می‌کند.

```
public void add(E[] a);
```

یک آرایه از نمونه‌های E را به CECollection مربوطه اضافه می‌کند.

```
public void clear();
```

CECollection مربوطه را خالی می‌کند.

```
public boolean containRef(E a);
```

در صورتی که refrence نمونه a در CECollection مربوطه وجود داشته باشد true و در غیر این صورت false برمی‌گرداند.

```
public boolean contain(E a);
```

در صورتی که نمونه‌ای برابر a (از لحاظ تابع equals()) درون CECollection مربوطه وجود داشته باشد true و در غیر این صورت false برمی‌گرداند.

```
public E remove(E a);
```

در صورتی که نمونه‌ای برابر a (از لحاظ تابع equals()) درون CECollection وجود داشته باشد آن را از CECollection پاک می‌کند و به عنوان خروجی تابع برمیگرداند(اگر چند نمونه وجود داشت همه آن‌ها را پاک می‌کند و یکی را به دلخواه برمی‌گرداند).

```
public boolean isEmpty();
```

اگر CECollection مربوطه خالی بود true و در غیر این صورت false برمیگرداند.

```
public int size();
```

تعداد نمونه‌های موجود در CECollection مربوطه را برمی‌گرداند.

```
public E[] toArray();
```

همه نمونه‌های درون CECollection مربوطه را به شکل آرایه‌ای از کلاس E برمی‌گرداند.

```
public class CEArrayList<E> implement CECollection<E>;
```

این کلاس در واقع همان CEArrayList جاواست که شما باید آن را پیاده سازی کنید. علاوه بر پیاده سازی کلاس‌هایی که به خاطر implement کردن CECollection، این کلاس باید پیاده‌سازی کند، توابع زیر را نیز برای این کلاس باید تعریف کنید.

توجه کنید که CEArrayList در واقع یک آرایه با طول متغیر است و هر تعداد عنصر دلخواه از نوع جنریک E را باید بتوان در آن جا داد. بنابراین در صورتی که در پیاده‌سازی از آرایه جاوا استفاده می‌کنید چون طول آرایه ثابت است باید تمهید لازم هنگامی که آرایه پر می‌شود را بیندیشید.

درصورتی که برای این کار از روش‌های بهینه مصرف حافظه و زمان استفاده کنید نمره امتیازی دریافت خواهید کرد.

توابع:

```
public E get(int index) throws CEArrayListIndexOutOfBoundException;
```

این تابع عضو indexام از CEArrayList مربوطه را برمی‌گرداند. در صورتی که index از اندازه CEArrayList مربوطه بزرگتر بود یک نمونه Exception از نوع CEArrayListIndexOutOfBoundException (این خطا را نیز شما باید تعریف کنید) باید throw شود.

```
public int indexOf(E a);
```

این تابع جایگاه (اندیس) اولین حضور عضو برابر a (از لحاظ تابع equal) در CEArrayList را برمی‌گرداند. در صورتی که a درون CEArrayList مربوطه وجود نداشت 1- برمی‌گرداند.

```
public void swap(int i, int j) throws CEArrayListIndexOutOfBoundException;
```

عضو i ام و j ام CEArrayList را بایکدیگر تعویض می‌کند. دی صورتی که i یا j بزرگتر از size آرایه بودند یک Exception از نوع CEArrayListIndexOutOfBoundException باید throw شود.

(کنجکاوی: مستقل از این سوال و تمرین!!!! به این فکر کنید که در جاوا چگونه می‌توانیم تابع swap برای دو متغیر از نوع primitive type بنویسیم!)

```
public class CESortedArrayList<E extends CEComparable> implement CECollection<E>;
```

این کلاس همه ویژگی‌های کلاس CEArrayList را داراست با این تفاوت همیشه نمونه ها به ترتیب sort شده در آن نگه‌داری می‌شوند. بنابر این CESortedArrayList تنها می‌تواند محتوی کلاس‌هایی باشد که interfaceمقایسه پذیری یعنی comparable را implement کرده باشند (تا ترتیب روی اعضا معنی پیدا کند!) همچنین به همین دلیل تابع swap برای آن معنی ندارد و بقیه توابع CEArrayList باید برای آن پیاده‌سازی شود.

public class CESet<E> implement CECollection

این کلاس باز نوع دیگری از CECollection است که همان طور که از نام آن مشخص است نمایشگر یک مجموعه است که به این معنی خواهد بود که عضو تکراری (از لحاظ تابع equal) را در خود نمی‌پذیرد و از هر عضو تنها یک نمونه نگه می‌دارد. بنابر این در صورتی که عضوی قبلا در CESet وجود داشت و باز به آن اضافه شد نباید تغییری در محتویات آن ایجاد شود.

این کلاس تنها همان توابع کلاس CECollection را پیاده‌سازی می‌کند و نیازی به پیاده‌سازی دیگر توابع نیست.

در صورتی که در پیاده‌سازی CESet از ساختارهای درختی و یا درهم‌سازی استفاده کنید نمره اضافی دریافت خواهید کرد.

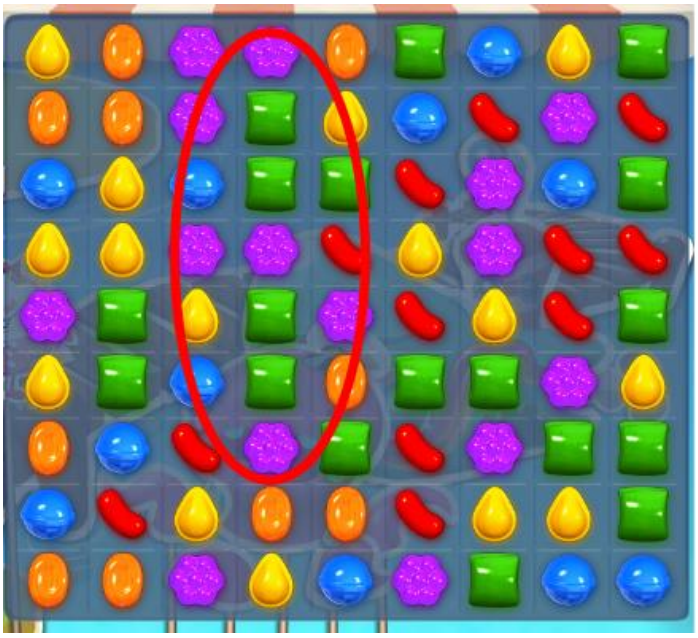
شهر شکلاتی

کندی بچه ی خیال بافی است که شکلات را خیلی دوست دارد. او همیشه در این آرزو بود که در مخفی شهر شکلاتی را پیدا کند .

او بعد از کنجکاوی فراوان بالاخره به آرزوی خود رسید و در شهر را پیدا کرد. در شهر شکلاتی بازی ای وجود داشت و هر فردی که در رقابت این شهر بیشترین امتیاز را کسب می کرد ، میتوانست شکلات دریافت کند. برای این که کندی بتواند در رقابت پیروز شود از تیم جاوا ی دانشکده در خواست کرد که این بازی را برای او پیدا سازی کنند تا قبل از رقابت بتواند تمرین کند.

بازی درخواست شده شامل یک جدول 9*9 بوده که در ردیف ها و ستون های آن ، شکلات قرار دارد. هرگاه که 3 یا بیشتر شکلات در این بازی در کنار هم قرار گرفتند ، این شکلات ها از خانه های جدول بازی حذف شده به کاربر امتیازی داده میشود و شکلات های جدید از بالای جدول ، جایگزین آن می شوند. بازی تا زمانی ادامه خواهد داشت که دیگر در جدول نتوان با جا به جا کردن شکلات ها ، امتیازی کسب کرد.

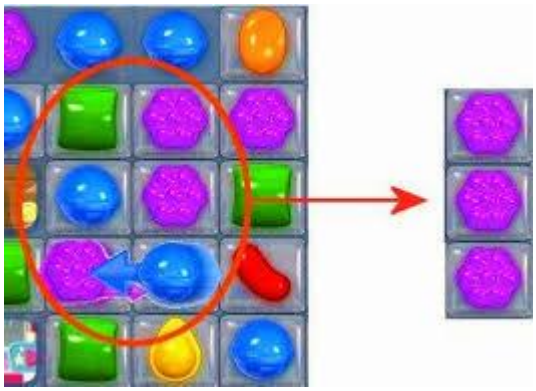
جدول بازی به صورت زیر خواهد بود :



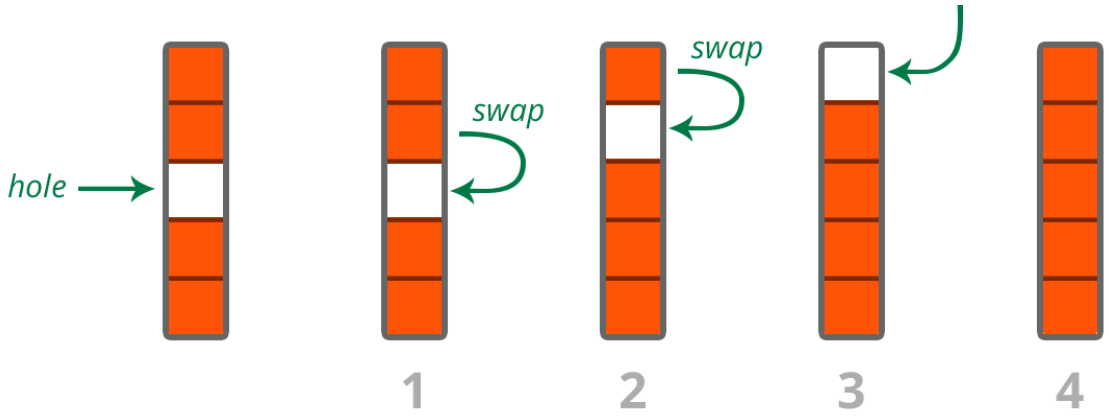
1) در این بازی 6 نوع شکلات وجود دارد و امتیاز همه ی شکلات ها باهم برابر می باشد.

برای حذف x شکلات	امتیاز
3	10
4	20
5	30
6	40
7	50
8	60
9	70

2) حذف شکلات ها زمانی امکان پذیر است که ، همه شکلات ها از یک نوع باشند.



3)جایگزینی به این شکلات ها زمانی که حذف می شوند به صورت زیر است :



فراموش نکنید که شکلات هایی که از بالای جدول وارد می شوند، به صورت رندم تولید می گردند.

4) زمانی که شکلات های جدید جایگزین قبلی ها خواهد شد ، باید درون جدول تمام خانه هایی که 3 یا بیشتر شکلات هم نوع در کنار هم قرار گرفته اند را حذف کرد.

5) زمانی که جا به جایی یک شکلات ، منجر به حذف شکلات های هم نوع خود می شود ، قبل از جایگزینی باید در جدول بررسی شود تا تمام خانه هایی که 3 یا بیشتر شکلات هم نوع در کنار هم قرار گرفته اند ، حذف گردند.

6) در ابتدای بازی باید نام کاربر از او گرفته شده و نامش در زمان بازی در کنار صفحه نمایش داده شود.

7) جمع امتیازات کاربر باید در صفحه ی بازی نمایش داده شود.

8) زمانی که بازی تمام شود ، صفحه ی بازی بسته شده و در صفحه ی دیگر نام و میزان امتیاز کاربر را نمایش دهد.

9) چنان چه در زمان جا به جایی یک شکلات ، عمل حذفی رخ ندهد ، شکلات به جای اول خود باز می گردد.

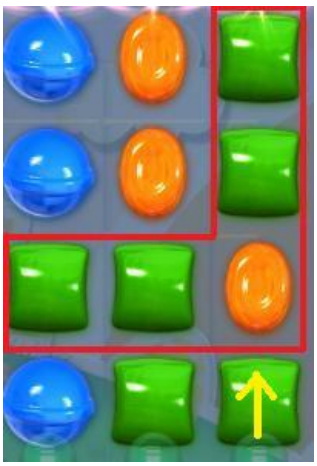
10) هر شکلات فقط می تواند با 4 خانه ی جدول جا به جا شود : بالایی ، پایینی ، سمت راست و سمت چپ .(خانه ی علامت گذاری شده با O با خانه هایی که علامت X دارند میتواند جا به جا شود و در صورتی که عمل حذف رخ ندهد به جای خود باز می گردد.)

11) جدول ابتدای بازی را میتوانید با الگوی دلخواه مورد نظر خود پر کنید . فقط یادتان باشد که در جدول ابتدایی نباید 3 یا بیشتر عنصر یکسان کنار هم باشند (یعنی در سطر یا ستون که منجر به حذف گردد.) .

جدول ابتدای بازی نباید به گونه ای باشد که امکان بازی برای کاربر وجود نداشته باشد.

12) اگر شکلات های همنوع هم در سطر و هم در ستون مشترک باشند ، باید سطر و ستون آن حذف گردد.

مانند :



13) برای راحتی در امتیاز دهی ، در هنگام حذف خانه ها ، تعداد خانه هایی که حذف میشوند را شمرده و مانند جدول امتیاز دهید.

14) آپشن های اضافه بر روی این بازی ، نمره ی اضافی خواهد داشت. برای ایده گرفتن نیز می توانید از سایت زیر استفاده کند:

Candy Crush saga