

فیو با ترد

- محدودیت زمان: 5 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- طراح سوال : محمد مهدی آورا
- سطح سوال : متوسط

در این سوال استفاده از ترد برای محاسبه جمله $n-1$ و $n-2$ به صورت همزمان الزامی است. (به صورت دستی چک میشود.) در این سوال از شما میخواهیم جمله های فیبوناچی را به کمک ترد محاسبه کنید یعنی مثلاً برای محاسبه جمله سوم شما باید جمله دوم و اول را به صورت همزمان محاسبه کنید.

*تابع فیبوناچی**، تابعی معروف است که نمایش *بازگشتی آن به این صورت است که هر جمله ی آن با توجه به دو جمله قبلی اش محاسبه می شود.

$$fib(0) = fib(1) = 1$$

$$fib(2) = 2$$

$$fib(n) = fib(n-1) + fib(n-2)$$

در جدول زیر نمونه هایی از ورودی و خروجی های این تابع داده شده است:

ورودی	خروجی
10	89
15	987

ورودی

ورودی تنها شامل یک خط است که در آن یک عدد طبیعی n آمده است.

$$n \leq 20$$

خروجی

خروجی برنامه‌ی شما باید شامل $fib(n)$ باشد.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

ورودی نمونه ۱

1

خروجی نمونه ۱

1

ورودی نمونه ۲

10

خروجی نمونه ۲

89

کارخانه تولید باقلوا

- محدودیت زمان: 1 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- طراح سوال : محمد مهدی آورا
- سطح سوال : متوسط

محمد در شرکت باقلوا پذی کار میکند. او مدیریت بخش تولید را در دست دارد.

داده های گذشته شرکت در فایلی که آدرس آن به کانسترامتر داده میشود ذخیره شده است و داده های امسال را نیز محمد به صورت دستی در کامپیوتر وارد میکند.

محمد در اینجا میخواهد میانگین تولید سال پیش و امسال را باهم مقایسه کند. در حال حاضر او میخواهد شروع کند به وارد کردن اطلاعات اما نميخواهد هنگامی که دارد اطلاعات را وارد میکند کامپیوتر بیکار باشد و میخواهد که کامپیوتر درحالی که دارد ورودی های او را میخواند میانگین داده های سال گذشته را نیز محاسبه کند.

بعد از اتمام ورودی ها باید میانگین ورودی ها را محاسبه کند و با میانگین سال های پیش مقایسه کند و اختلاف آن دو را در خروجی چاپ کند. شما میتوانید پروژه اولیه را از [این لینک](#) دانلود کنید.

کلاس GetDats

در این کلاس شما باید با خواندن ورودی ها از طریق اسکنر خواسته شده میانگین سال مورد نظر را حساب کنید.

متد getAverage

این متد مقدار میانگین را خروجی میدهد و اگر میانگین هنوز حساب نشده باشد باید 1- را در خروجی دهد.

متد isOutputReady

خروجی این متد میگوید که آیا با ران شدن ترد خروجی حساب شده است یا خیر پس اگر ترد اجرا شده باشد باید true خروجی دهد در غیر این صورت باید false خروجی دهد.

کلاس ReadData

در این کلاس شما باید با خواندن ورودی ها از فایل خواسته شده میانگین سال مورد نظر را حساب کنید.

متد getAverage

این متد مقدار میانگین را خروجی میدهد و اگر میانگین هنوز حساب نشده باشد باید 1- را در خروجی دهد.

متد isOutputReady

خروجی این متد میگوید که آیا با ران شدن ترد خروجی حساب شده است یا خیر پس اگر ترد اجرا شده باشد باید true خروجی دهد در غیر این صورت باید false خروجی دهد.

ورودی

محمد تعداد روز هایی که شرکت فعال بوده است را فراموش کرده است و همینطوری اعداد را وارد میکند تا زمانی که ورودی ها تمام شود. و سپس شما باید میانگین اعداد (n) را محاسبه کنید. در فایلی که آدرس آن به کانستراکتر داده شده نیز در هر خط یک عدد که نشان دهنده میزان تولید آن روز است آمده است و شما باید میانگین اعداد (m) را محاسبه کنید. نکته: ورودی ها میتوانند اعشاری باشند.

خروجی

در خروجی برنامه شما باید $n-m$ را با دو رقم اعشار چاپ کنید.

مثال

ورودی نمونه ۱

3
3
4
2

فایل نمونه ۱

1
2
3

خروجی نمونه ۱

1.0000

استخراج

- نام طراح: سپهر ابراهیمی
- سطح سوال: ساده

روش‌های مختلفی برای فشرده سازی داده‌ها وجود دارد. در این سوال (و دو سوال بعدی) با یک نمونه‌ی خیلی ساده آشنا می‌شوید که برای فشرده کردن اطلاعات چند فیلم از آن استفاده شده است.

ابتدا پروژه‌ی اولیه را از [این لینک](#) دانلود کنید. در این فایل، صرفاً یک کلاس با نام `Extractor` و یک تابع با نام `extract` خواهید دید. در امضای تابع (نام، ورودی‌ها، خروجی و ویژگی‌هایی مثل `static` بودن) و کلاس نباید چیزی تغییر بکند، اما می‌توانید توابع دیگری به تشخیص خودتان به این کلاس اضافه کنید. در نهایت نسخه‌ی تکمیل شده‌ی همین فایل را آپلود خواهید کرد.

نحوه‌ی استخراج را با یک ورودی و خروجی نمونه بررسی می‌کنیم:

ورودی:

ورودی شما یک فایل است به نام `in.txt` که در آدرس `baseAddress` که به تابع پاس داده خواهد شد، قرار دارد. به عنوان مثال، اگر فایل ما در درایو `C` در یک پوشه به اسم `folder` قرار داشته باشد، آدرس فایل `C:\folder\in.txt` خواهد بود و مقدار `baseAddress` برابر:

`C:\folder\`

خواهد بود. در این فایل، هر خط نشان‌دهنده‌ی یک فیلم است. به عنوان مثال:

```
first movie;link to first movie;so high;me
```

این خط یعنی نام فیلم `first movie` است، لینکی به صفحه‌ی فیلم در اینترنت وجود دارد که `link to` `first movie` است، نمره‌ی این فیلم `so high` است و کارگردانش `me`. فعلاً هر فیلمی که در این فایل

ذخیره شده، دقیقا همین ۴ ویژگی را خواهد داشت و این ویژگی‌ها با ؛ از هم جدا شده‌اند. در هر کدام از این ویژگی‌ها نیز ؛ وجود نخواهد داشت. هر فیلم در یک خط جای خواهد گرفت و در هر خط یک فیلم وجود دارد.

خروجی:

شما باید در کنار همین فایل (یعنی در `baseAddress`) یک پوشه به نام `movies` بسازید. سپس به ازای هر فیلم، یک فایل به نام فیلم (یعنی اگر نام فیلم `first movie` است، نام فایل شما `first movie.txt` است) خواهید ساخت. توجه کنید بعضی کارکترها در نام فایل مجاز نیستند ولی ممکن است در نام فیلم موجود باشند (مثلا : ، می‌توانید یک فایل درست کنید و سعی کنید هنگام عوض کردن نامش، یک کارکتر غیرمجاز وارد کنید تا در یک دیلوگ به شما تمامی این کارکترها را نمایش دهد. برای راحتی کار ، این کارکترها رو اینجا مینویسم:

\
/
:
\
*
?
"
<
>
|

دی (برای این موارد، کافیهست این کارکترها از نام فایل حذف شوند. مثلا اگر نام فیلم `Top Gun` :
`Maverick` است، نام فایل `Top Gun Maverick` خواهد بود. در هر فایل، اطلاعات خط به خط نوشته می‌شوند. به داده‌های مورد انتظار در فایل `first movie.txt` دقت کنید:

Name: first movie
Link: link to first movie
Rating: so high
Director: me

در نهایت اگر `baseAddress` ما همان

`C:\folder\`

باشد، بعد از اجرای برنامه‌ی شما باید یک پوشه به اسم `movies` با یک فایل به اسم `first movie.txt` در آن، داخل آدرس بالا ایجاد شده باشد. یعنی:

`C:\folder\movies\first movie.txt`

فشرده سازی

- نام طراح: سپهر ابراهیمی
- سطح سوال: ساده-فایل

این بار قرار است با دریافت پوشه‌ی movies، فایل فشرده‌ی مد نظر را تولید کنید:

ابتدا پروژه‌ی اولیه را از این لینک دانلود کنید. در این فایل، صرفاً یک کلاس با نام Compressor و یک تابع با نام compress خواهید دید. در امضای تابع (نام، ورودی‌ها، خروجی و ویژگی‌هایی مثل static بودن) و کلاس نباید چیزی تغییر بکند، اما می‌توانید توابع دیگری به تشخیص خودتان به این کلاس اضافه کنید. در نهایت نسخه‌ی تکمیل شده‌ی همین فایل را آپلود خواهید کرد.

نحوه‌ی فشرده سازی را با یک ورودی و خروجی نمونه بررسی می‌کنیم:

ورودی:

ورودی شما یک پوشه است به نام movies که در آدرس baseAddress که به تابع پاس داده خواهد شد، قرار دارد. به عنوان مثال، اگر پوشه‌ی ما در درایو C در یک پوشه به اسم folder قرار داشته باشد، آدرس پوشه C:\folder\movies خواهد بود و مقدار baseAddress برابر:

C:\folder\

خواهد بود. در این پوشه، هر فایل نشان‌دهنده‌ی یک فیلم است که مانند همان فایل‌هایی که در سوال قبلی ایجاد کرده‌اید، خواهند بود. به عنوان مثال، اگر همان فیلم سوال قبلی را در نظر بگیریم، در این پوشه باید یک فایل با نام first movie.txt ببینیم که محتویات آن به شکل زیر است:

Name: first movie

Link: link to first movie

Rating: so high

Director: me

دقت کنید که همچنان دقیقا همین ۴ ویژگی در فایل‌های ورودی وجود خواهند داشت. حواستان به تفاوت احتمالی نام فایل با نام فیلم نیز باشد D: در ضمن، تابع Files.listFiles نیز برای این سوال مناسب است (=)

خروجی:

شما باید در کنار همین پوشه (یعنی در baseAddress) یک فایل به نام out.txt بسازید که فشرده‌ای از همه‌ی فیلم‌های موجود در پوشه‌ی movies را در خود دارد. نحوه‌ی فشرده کردن نیز مانند سوال قبل است. یعنی اگر تنها فایل موجود در پوشه‌ی movies همان first movie.txt باشد، فایل out.txt شما مانند زیر خواهد بود:

```
first movie;link to first movie;so high;me
```

در نهایت اگر baseAddress ما همان

C:\folder\

باشد، بعد از اجرای برنامه‌ی شما باید یک فایل به اسم out.txt در آن، داخل آدرس بالا ایجاد شده باشد. یعنی:

C:\folder\out.txt

جعبه ابزار

- نام طراح: سپهر ابراهیمی
- سطح سوال: متوسط

حال می‌خواهیم در یک کلاس، هر دو عمل فشرده سازی و استخراج را انجام دهیم، اما با یک تفاوت جزئی (: ابتدا پروژه‌ی اولیه را از [این لینک](#) دانلود کنید. در این فایل، صرفاً یک کلاس با نام Utility و دوتابع با نام‌های extract و compress خواهید دید. در امضای توابع (نام، ورودی‌ها، خروجی و ویژگی‌هایی مثل static بودن) و کلاس نباید چیزی تغییر بکند، اما می‌توانید توابع دیگری به تشخیص خودتان به این کلاس اضافه کنید. در نهایت نسخه‌ی تکمیل شده‌ی همین فایل را آپلود خواهید کرد.

ورودی و خروجی:

ورودی‌ها و خروجی‌ها دقیقاً مانند دو سوال قبل است (هر تابع مشابه با سوال مرتبطش). تنها تفاوت این است که تعداد ویژگی‌های هر فیلم از پیش مشخص نیست. برای نمونه، ممکن است یک فیلم تنها نام و لینک داشته باشد، در حالی که دیگری علاوه بر ۴ ویژگی سوالات قبل، نویسنده(ها) را نیز ذکر کرده باشد. تضمین میشود که برای هر ورودی، این تعداد تغییر نمیکند (یعنی در یک فایل in.txt ، همه‌ی فیلم‌ها تعداد و نوع یکسانی از ویژگی‌ها را دارند)، اما این تعداد بین ورودی‌های مختلف یکسان نیست. برای راحتی کار، اگر در حال استخراج باشید، اولین خط از فایل همیشه برای ذکر کردن نوع این ویژگی‌ها خواهد بود. به عنوان مثال، اگر از شما بخواهیم همان اطلاعات مثال سوالات قبل را استخراج کنید، فایل ورودی به این شکل خواهد بود:

```
Name;Link;Rating;Director
first movie;link to first movie;so high;me
```

در خط اول به ترتیب تیتل هر ویژگی خواهد آمد، و در خطوط بعدی برای هر فیلم، این ویژگی‌ها مقدار داده خواهند شد. تضمین می‌شود نام هیچ کدام از این ویژگی‌ها در خود ؛ یا : نخواهد داشت (پس

می‌توانید با استفاده از : هنگام فشرده سازی، نام هر ویژگی را از مقدار جلویش تمییز دهید. حواستان باشد که این خط اول را شما نیز باید در زمان فشرده سازی در فایل خروجی خود قرار دهید D: (

رشته های دوست داشتنی

مباحث تحت پوشش:

- ساختمان داده set
- کار اولیه با رشته
- ساختمان داده map
- کالکشن های unmodifiable

سطح سوال: ساده طراح: روزبه شریفنسب

علی بعضی رشته ها را دوست دارد و بعضی از رشته ها را دوست ندارد.

اون از شما می خواهد یک Set از رشته ها به عنوان ورودی بگیرید و دو خروجی برگردانید:

- یک مجموعه شامل همه رشته هایی که از set ورودی دوست دارد.
- و همچنین یک مجموعه شامل همه رشته هایی که علی دوست ندارد.

کد کلی کلاس به شکل زیر می شود:

 Splitter.java

```
1  import java.util.*;
2
3  public class Splitter {
4
5      public Splitter(final Set<String> set) {
6      }
7
8      public Set<String> goods() {
9          return null;
10     }
11
12     public Set<String> bads() {
```

```

13         return null;
14     }
15
16     public Map<String, Set<String>> split() {
17         // keys: "goods", "bads"
18         return null;
19     }
20 }

```

همانطور که در کد بالا می‌بینید یک تابع split هم وجود دارد که کارش این است که یک map برگرداند. در واقع این map شامل دو داده با کلید های goods و bads است که دو مجموعه خوبها و بدها را برمی‌گرداند. از شما می‌خواهیم که کد خواسته شده را پیاده سازی کنید.

نکات پیاده سازی

- برنامه شما نباید با تغییر دادن set و map های خروجی، خروجی اش تغییر کند بلکه باید اکسپشن پرتاب کند. مثلا این کد باید اکسپشن دهد:

```

1 Set<String> set = new HashSet<>();
2 set.add("a");
3 set.add("b");
4 set.add("c");
5 Splitter splitter = new Splitter(set);
6 Set<String> goods = splitter.goods();
7 goods.clear(); // exception

```

- می‌توانید unmodifiable ها سرچ کنید.

رشته‌هایی که علی دوست دارد

علی در انتخاب رشته هایی که دوست دارد کمی سخت‌گیر است!

برای اینکه بفهمیم علی یک رشته را دوست دارد یا نه باید مراحل زیر را طی کنیم:

۱. رشته ورودی را تبدیل به ROT13 کنیم.

۲. تعداد کارکترهای a تا p (شامل خود a, p) در رشته ROT13 شده فرد باشد.
۳. درمورد ROT13 می‌توانید [اینجا](#) بخوانید.

روزنامه‌نگار جوان

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- طراح سوال: مبینا شهبازی

روزنامه نگار جوانی به تازگی در یک شرکت استخدام شده؛ رئیس او آدم دقیق و سخت گیری است و هر بار از کارمندانش می‌خواهد اطلاعات خاصی را از متن خبرها استخراج کنند. روزنامه نگار ما که بی تجربه است از شما می‌خواهد با تکمیل برنامه زیر به او کمک کنید. در این سوال، یک کلاس Main وجود دارد که دارای ویژگی های filepath و text است. باید با استفاده از filepath ویژگی text را در کانستراکتور مقداردهی کنید. همچنین باید توابع زیر را پیاده سازی کنید. پروژه اولیه را از این لینک دانلود کنید. تضمین میشود در محتوای فایل موردنظر :

- جملات شامل حروف، عدد، "." و "," هستند.
- تمام جملات با کلمه ای با حرف بزرگ شروع میشوند ولی اسم خاص نیستند.

۱. متد getWords :

این تابع لیستی از تمام کلمات text برمی‌گرداند.(با تکرار کلمات و حفظ ترتیب آنها)

توجه داشته باشید ارقام و اعداد کلمه محسوب نمیشوند.

۲. متد MBNIncludes :

این تابع یک TreeMap از کلمات دارای حروف (m, b, n, M, B, N) برمی‌گرداند. key خود کلمات و value تعداد تکرار آن کلمه در متن خواهد بود.

۳. متد getProperNouns :

در این تابع باید با پیمایش متن اسامی خاص را پیدا کنید. درنهایت آنها را به صورت یک arraylist (اما بدون تکرار) برگردانید.

۴. متد MAverage :

این تابع باید لیستی از کلماتی که میانگین وزن آنها برابر با m باشد را برگرداند. (با تکرار کلمات)

- میانگین وزن کلمات: در این سوال منظور از میانگین وزن کلمه مجموع حروف به صورت `int` تقسیم بر تعداد حروف است. برای مثال میانگین در هر بخش مشخص شده:

```
abc -> b
XYZ -> Y
abcd -> invalid
```

۵. متد getFrequency :

این تابع لیستی از تمام کلمات متن به ترتیب تکرار آنها در متن برمی‌گرداند (به صورت صعودی). اگر تعداد دفعات تکرار چندین کلمه باهم برابر باشد ترتیب قرارگیری آنها در لیست بر اساس حروف الفبا خواهد بود. (ابتدا A تا Z و سپس a تا z)

در این بخش هم ارقام و اعداد کلمه محسوب نمی‌شوند.

مثال

```
FileWriter fr=new FileWriter("src/sample.txt");
fr.write("My words fly up, my thoughts remain below. Words without thoughts never to l
fr.flush();
Main m=new Main("src/sample.txt");
System.out.println(m.getWords());
System.out.println(m.getProperNouns());
```

اگر کد بالا را در `main` تست کنید، انتظار می‌رود خروجی به شکل مقابل باشد:

[My, words, fly, up, my, thoughts, remain, below, Words, without, thoughts, never, to,

[]

مطابق صورت سوال کلاس Main را تکمیل و آپلود کنید.