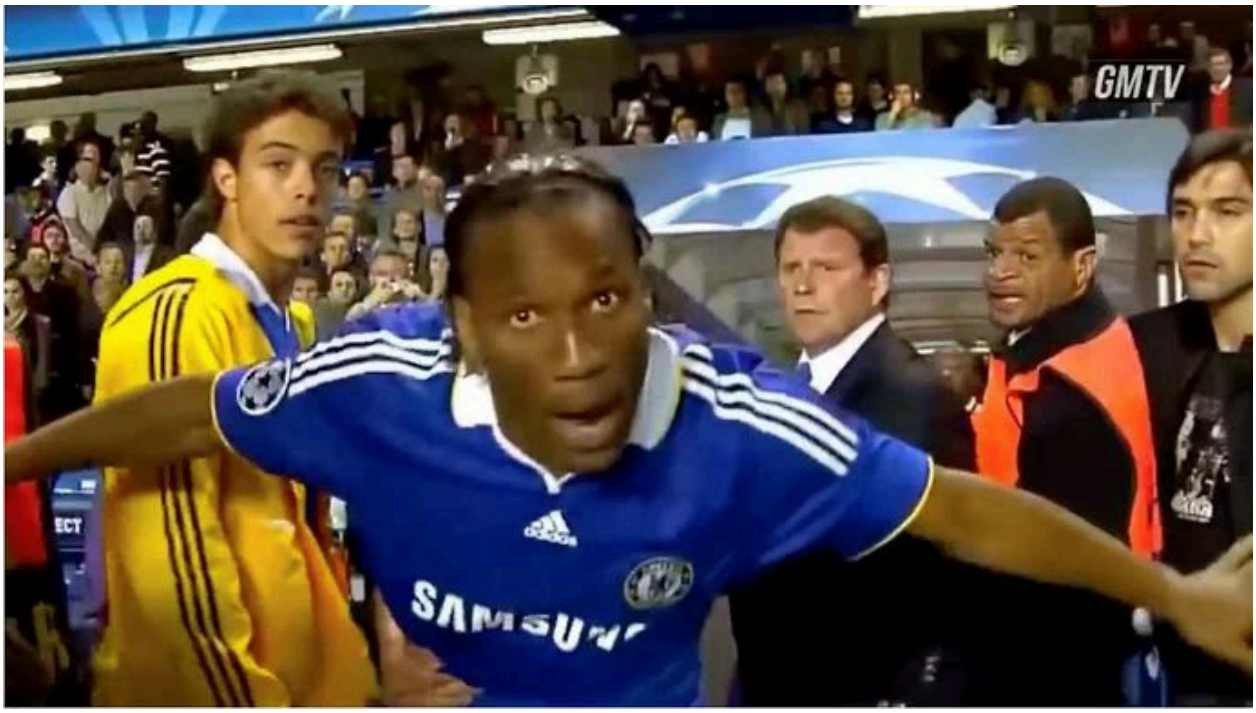
کاراپشن

داستان از کجا شروع شد؟

سالیان سال است که فدراسیون بین‌المللی فوتبال (FIFA) مانند خدایانی ظالم در تمام اتفاقات دنیای فوتبال دخالت می‌کنند. این فرایند در دهه اخیر به طرز فاجعه باری از کنترل خارج شد به طوری که تیمی بدون افتخار و هویت، موفق شد دارای 5 بار قهرمانی لیگ قهرمانان اروپا (Champions League) شود و حتی توپ طلا (ballon d'or) که ملاکی برای نشان دادن فوتبالیست‌های افسانه‌ای بود 8 بار در دهه اخیر به سرقت رسید.

اولین شک و شبهه‌ها بر وجود دستی بالاتر در این بازی‌ها پس از نیمه‌نهایی 2009 لیگ قهرمانان شکل گرفت. بازی‌ای که داور آن حتی سال‌ها پس از گذشت آن بازی مورد تهدید هواداران تیم بازنده قرار داشت و بخاطر خیانتی که به دنیای فوتبال کرد، شب‌ها خواب راحت نداشت.



سال‌ها می‌گذرد و بازیکنان و باشگاه‌های فوتبالی، ناچار با شرایط کنار می‌آیند و این خفت و خواری را به جان می‌خرند که شاید سالی برسد که خدایان فوتبال، آنها را به‌عنوان قهرمان انتخاب کند.

در سال 2024 بازیکنی جوان و تازه نفس به این میدان جنگ وارد می‌شود و از بدو ورود خود در معرض بی‌عدالتی‌ها و نابرابری‌های بی‌شماری قرار می‌گیرد.

اون هنوز جوانی خام است و نمی‌تواند آرام بگیرد و با شرایط کنار بیاید. روزها می‌گذرد و هر هفته شاهد مورد ظلم قرار گرفتن تیمش در زمین می‌شود. روزی از روزها پس از 4 باخت متوالی که همه آنها توسط فیفا برنامه‌ریزی شده بود دیگر تصمیم می‌گیرد که دربرابر این ظلم و بی‌عدالتی قیام کند. حتی در بازی آخرش موفق به زدن 3 گل برای تیمش شد و هتریکی باشکوه به نمایش گذاشت، اما همچنان نتوانست رقیب script فوتبال شود.

او می‌خواهد به سرور های فیفا نفوذ کرده و اسرار مخفی فوتبال را پیدا کند. برای اینکار لازم است پسورد رئیس فیفا را پیدا کند. او N نفر را مأمور می‌کند که با استفاده از الگوریتم هش SHA-256 رمز عبور را از روی هش آن پیدا کنند. آن‌ها یک لیست از رمزهای احتمالی دارند و می‌خواهند با بالاترین کارایی ممکن آن را جستجو کنند. رمزهای احتمالی در یک فایل با نام `passwords.txt` ذخیره شده است و هر رمزی که مقدار هش آن با هش داده شده برابر باشد، رمز صحیح می‌باشد.

ذخیره رمزهای کاربران به صورت Plain text در پایگاه‌های داده بسیار ناامن است. به همین منظور، از عملیاتی به نام Hashing(هش کردن) برای احراز هویت کاربران استفاده می‌کنیم.

هش کردن (Hashing) فرایندی است که طی آن یک آرایه به طول دلخواه را به یک آرایه با طول ثابت تبدیل می‌شود. این رشته خروجی، هش نامیده می‌شود. از تابع هش برای مقاصد مختلفی مانند رمزنگاری استفاده می‌شود. این فرایند باید برگشت‌ناپذیر باشد تا از روی خروجی آن نتوان به رشته اصلی دست پیدا کرد.

در سامانه‌های احراز هویت، نام کاربری در کنار هش رمز عبور (به جای خود رمز عبور) ذخیره می‌شود، به این شکل هنگامی که سیستم بخواهد رمز کاربر را تایید کند، هش آن را محاسبه می‌کند و با هش‌ای که هنگام ثبت نام کاربر محاسبه کرده بود، مقایسه می‌کند. با انجام این فرایند حتی در صورت پخش شدن اطلاعات پایگاه داده، رمزها به صورت مستقیم فاش نخواهند شد.

الگوریتم‌های هش کردن زیادی وجود دارد که در این سوال به الگوریتم SHA-2 (به طور خاص تر، SHA-256) می‌پردازیم. برای آشنایی بیشتر با توابع هش رمزنگاری، میتوانید به این لینک و با خانواده SHA-2 به این لینک مراجعه کنید. در صورت علاقه مندی بیشتر، در اینجا می‌توانید با نحوه کار الگوریتم SHA-2 بیشتر آشنا شوید.

برای استفاده از تابع هش SHA-256، میتوانید از کلاس زیر در جاوا استفاده کنید:

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.nio.charset.StandardCharsets;

public class CryptoHash {
    public static String hashString(String text) {
        try {
```

```
            MessageDigest digest = MessageDigest.getInstance("SHA-2
            byte[] encodedhash = digest.digest(text.getBytes(Standa
                StringBuilder hexString = new StringBuilder(2 * enc
            for (byte b : encodedhash) {
                String hex = String.format("%02x", b);
                hexString.append(hex);
            }

            return hexString.toString();

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

ورودی

در خط اول ورودی به شما یک هش ۳۲ بایتی با نمایش Hexadecimal داده می شود. همچنین در خط دوم ورودی به شما عدد N داده می‌شود. همچنین فایل `passwords.txt` را در انتهای سوال می‌توانید دریافت کنید. تضمین می‌شود هش داده شده از N رمز‌عبور آخر نیست.

▼ *برای آشنایی بیشتر*

برای تمرین، می‌توانید تمام رمزهای فایل را کنار یکدیگر **بدون وایت اسپیس** گذاشته و هش آن را بررسی کنید که با مقدار زیر یکی شده است! (طبیعتا این بخش الزامی نیست و نمره‌ای ندارد و هدف آن صرفاً آشنایی بیشتر دانشجو با عملیات هش‌کردن است).

```
609355ff03275dd9d9c1144b585b03b25526d188a16d85d83154df93d2f0e049
```

نحوه اجرا

با توجه به ماهیت سوال، **باید تمام شرایط زیر رعایت شود** تا جواب یکتا به دست بیاید.

- به تعداد N ترد ثابت داریم که هم‌زمان اجرا می‌شوند.
- در هر سری، باید **کار هر N ترد تمام شود** تا سراغ N ترد بعدی برویم.
- لیست رمزهای عبور به صورت صف (queue) است. (یعنی به ترتیب از اول لیست رمزهای عبور تست شوند)
- هر ترد پس از تمام شدن بررسی یک رمز عبور، به‌صورت خودکار رمز عبور بعدی را از صف می‌گیرد.
- عملیات تا زمانی ادامه دارد که یا رمز عبور صحیح پیدا شود، یا همه رمزهای عبور بررسی شده باشند.

پیشنهاد : در مورد Executors و AtomicReference تحقیق کنید.

تعریف ویژگی خاص

در طول بررسی رمزهای عبور، تردها هش SHA-256 هر رمز را محاسبه می‌کنند. اگر حداقل یکی از ۳ رقم اول (از سمت چپ) هش در بازه 0 تا 6 باشد، آن پسورد ویژگی خاص دارد. تعداد کل پسوردهای با ویژگی خاص با در نظر گرفتن و حل **race condition** ها باید به ما برگردانده شود.

توابع مورد نیاز در کلاس PasswordBreaker

تابع	توضیح
void setTargetHash(String hash)	هش رمز هدف را تعیین می‌کند.
void loadPasswords(File file)	لیست رمزهای احتمالی را از فایل بارگذاری می‌کند.
void startCracking(int numThreads)	عملیات کرک را با N ترد موازی آغاز می‌کند.
String getFoundPassword	در صورت یافتن رمز صحیح، آن را برمی‌گرداند. در غیر این صورت null بر میگرداند.
int getSpecialHashCount	تعداد پسوردهایی که ویژگی خاص دارند.

خروجی

در ابتدا، تعداد رمزهای عبور با ویژگی خاص باید چاپ شود. سپس، در صورتی که رمز عبور صحیح پیدا نشد، جمله `NOT FOUND` چاپ می شود. در غیر این صورت، رمز عبور صحیح باید چاپ شود. تضمین می‌شود که کاراکتر اسپیس در کلمات موجود در فایل وجود ندارد.

مثال

اگر ورودی زیر به شما داده شود:

```
00f7ef114a768fce3089a2204624984c6f24646a0fd00acc5799854d628aabc1
5
```

خروجی باید به شکل زیر باشد:

```
12
k#iW*ipOletab^Le%@
```

نکته: اگر به صورت Single Thread بررسی را انجام می‌دادیم، هنگامی که به رمز مدنظر می‌رسیم تنها ۱۰ رمز با ویژگی خاص پیدا می‌کنیم اما از آن‌جایی که قرار شد به صورت دسته‌های ۵ تایی رمزها را بررسی کنیم، دو رمز دیگر بعد از رمز شماره ۱۳ هم رمز عبور صحیح خواهند بود که با استفاده از دسته‌های ۵ تایی پیدا می‌شوند.

ساختار پروژه و نحوهٔ داوری

ساختار پروژهٔ شما باید به صورت زیر باشد:

```
Main.java
PasswordBreaker.java
CryptoHash.java
passwords.txt
```

فایل passwords.txt را برای تست خودتان باید در کنار پروژهٔ خود قرار دهید: **دریافت فایل**

توجه: نیازی به قرار دادن به فایل passwords.txt داخل فایل زیپ ارسالی نیست. در سامانهٔ داوری به‌طور خودکار این فایل کنار پروژه قرار می‌گیرد.

پیشنهاد: می‌توانید یک کد python/bash بنویسید که به ازای یک hash ثابت، و تعداد ترد (N) متغیر، نمودار زمان را رسم کرده یا مقادیر زمان را چاپ کند؛ و برآورد کنید که مقدار N بهینه برای ترکیب این مسئله و سیستم شما چه عددی است.