



## تمرین - سری اول

به جز ۲ تمرین آخر، باقی تمرینات باید به شکل اسکریپت‌های `bash` نوشته شوند. برای هر تمرین لطفا توضیح کامل بدهید که دستورات شما هر کدام چه کاری انجام می‌دهند.

**مساله ۱.** اسکریپتی بنویسید که به عنوان آرگومان ورودی آدرس یک دایرکتوری را بگیرد و در صورتی که دایرکتوری موجود نبود خطای مناسب را به کاربر نمایش داد و با کد ۱ از برنامه خارج شود. در صورتی که آدرس درست بود، طول اسم فایل‌های این دایرکتوری را بدون پسوند در نظر بگیرد و ماکزیمم این اعداد را در خروجی چاپ کند.

**مساله ۲.** یک متغیر تعریف شده داریم به نام `RANDOM` که با هر بار صدا زده شدن یک عدد تصادفی بین ۰ و ۳۲۷۶۷ برمی‌گرداند.

**الف)** اسکریپتی بنویسید که راستی آزمایی این باشد که عددی که به این شکل تولید می‌شود از یک توزیع یکنواخت می‌آید.

**ب)** فرض کنید می‌خواهید عدد تصادفی در بازه ۰ تا  $n$  تولید کنید. با استفاده از متغیر فوق تابعی بنویسید که عدد  $n$  را به عنوان ورودی بگیرد و یک عدد تصادفی برگرداند. راستی آزمایی را برای تابع خود نیز انجام دهید.

**مساله ۳.** در `/proc/cpuinfo` اطلاعات مربوط به پردازنده‌ی سیستم را می‌توانید ببینید. با استفاده از آن اسکریپتی بنویسید که تعداد پردازنده‌های سیستم شما را نمایش دهد.

**مساله ۴.** اسکریپتی بنویسید که از آدرس

[https://en.wikipedia.org/wiki/Bash\\_\(Unix\\_shell\)](https://en.wikipedia.org/wiki/Bash_(Unix_shell))

شروع کند و تمام صفحات تا عمق ۲ را پیش رفته و تعداد تکرار هر کلمه را حساب کند.



مساله ۵. یک ریپازیتوری گیت ایجاد کنید که برنچ‌های زیر را داشته باشد:

```
fib ۱ ۲
fib ۲ ۳
fib ۴ ۵
...
fib ۲n ۲n + ۱
...
fib ۲۰۴۸ ۲۰۴۹
```

در هر برنچ، هزار کامیت می‌خواهیم داشته باشیم که پیام کامیت  $i$ ام

added the  $i$ th number

باشد و در آن کامیت صرفاً یک فایل با اسم

$i$ .txt

موجود باشد که محتوای آن عدد  $i$ ام یک دنباله‌ی فیبوناچی است که دو عدد اول دنباله از اسم برنچ گرفته شده است. مثلاً در برنچ `fib_۴_۵` چند کامیت اول این‌ها خواهند بود:

added the 1st number:

1.txt: 4

added the 2nd number:

2.txt: 5

added the 3rd number:

3.txt: 9

added the 4th number:

4.txt: 14

در مثال بالا، توجه کنید که  $i$ th برای اعداد یک، دو و سه حالت استثنایی دارد.

مساله ۶. در مورد *availability* و *consistency* در کلاس صحبت کردیم و با توجه به قضیه CAP نمی‌توان همزمان هر دوی این موارد را به طور کامل داشت. در هر یک از نمونه‌های زیر توضیح دهید که انتخاب کدامیک از این موارد در شرایط اضطراری مناسب‌تر است:

- اگر با نرم‌افزار اسپاتیفای آشنا باشید، لیست‌هایی پیشنهادی به هر کاربر ارائه می‌کند. برای ساخته شدن این لیست‌ها نیاز به داشتن اطلاعات کاربران در زمان‌های گذشته وجود دارد و این لیست‌ها براساس این اطلاعات ساخته می‌شود. با توجه به حجم بالای این اطلاعات از سیستم‌های توزیع شده برای نگهداری و محاسبات روی آن‌ها استفاده می‌شود. در این حالت اولویت شما چیست؟

- در شتاب‌دهنده‌ی سرن، حجم اطلاعات تولید شده بسیار بالاست و برای نگهداری اطلاعات از سیستم‌های توزیع شده استفاده می‌شود. این اطلاعات برای بررسی نظریه‌های فیزیکی کاربرد دارد. در این مورد برای بررسی این نظریه‌ها تصمیم شما چیست؟



مسئله ۷. در مورد کش توزیع شده در کلاس صحبت کردیم و برای بحث *scalability* از رپلیکیشن‌هایی برای خواندن استفاده شده به همراه لیدر. حال فرض کنید که برای رپلیکیشن بخواهیم از همان نودهایی که به عنوان سرورهای اصلی هستند و برای توزیع کش از آن‌ها استفاده کرده‌ایم استفاده کنیم. این کار چطور ممکن است، چه مزایایی دارد و چه مشکلاتی ممکن است پیش آورد؟