

سوال ۱:

۷ خط اول نکته خاصی ندارد، فقط در if توجه شود که:

[-d FILE]	True if FILE exists and is a directory.
[-e FILE]	True if FILE exists.

در خط آخر هم توجه شود که e- به این خاطر قرار داده شده تا n\ کار کند:

Note: The '-e' option in Linux acts as an interpretation of escaped characters that are backslashed.

در بقیه سوالات از دستورات عادی bash بدون استفاده از آپشن خاصی استفاده شده است و کاری که هر کدام انجام می‌دهند هم با توجه به نامگذاری‌ها معلوم است.

سوال ۲:

(الف)

خود کد نکته خاصی ندارد، در ۵ خط اول یک آرایه ساخته شده و سپس ۱۰۰۰۰۰ عدد تصادفی در آن جای گرفته است. سپس فاصله ۰ تا ۳۲۷۶۷ به بازه‌هایی به طول ۵۰ تقسیم شده و سپس χ^2 و زیگما محاسبه شده. (توضیحات در ادامه)

توضیحات روش chi-square مورد استفاده:

فاصله ۰ تا ۳۲۷۶۷ را به تعدادی بازه هم‌طول (در اینجا ۵۰) تقسیم می‌کنیم. از ۱۰۰۰۰۰ عددی که تولید می‌شوند با توجه به اینکه ۶۵۵ بازه داریم امید ریاضی تعداد ارقام هر بازه تقریباً ۱۵۳ است. حالا برای هر بازه مقدار χ^2 را حساب می‌کنیم و آنگاه مجموع (زیگما) χ^2 ها را به دست می‌آوریم. نحوه محاسبه χ^2 برای هر بازه:

$$\chi^2 = \frac{(ObservedValue - ExpectedValue)^2}{ExpectedValue}$$

در بهترین حالت برای هر بازه $ObservedValue = ExpectedValue$ پس مجموع χ^2 ها صفر است. در بدترین حالت تمام اعداد در یک بازه قرار می‌گیرند پس مجموع χ^2 ها برابر است با:

$$(654 \times ExpectedValue) + \frac{(100000 - 153)^2}{153} = 65259692$$

حال کد ما با این ارقام چندبار اجرا شده و مقادیری بین ۴۵۰ و ۵۵۰ برای زیگما به دست آورده که این ارقام در بازه ۰ تا ۶۵۲۵۹۶۹۲ به وضوح بسیار به صفر نزدیک‌تر هستند که از این مطلب نتیجه می‌شود که اعدادی که متغیر RANDOM تولید می‌کند تا حد خیلی خوبی از توزیع یکنواخت می‌آیند.

(ب)

در این قسمت هم از همان ایده‌های قسمت الف استفاده می‌شود با این تفاوت که مقدار امید ریاضی با توجه به ورودی حساب می‌شود و در انتها بهترین و بدترین مقدار زیگما هم نمایش داده می‌شوند، پس از هر بار ران کردن کد ملاحظه خواهید که با باز هم مقادیر زیگما نسبت به بازه بسیار به صفر نزدیک هستند و این نشان می‌دهد که اعداد تولیدی تا حدی خیلی خوبی از توزیع یکنواخت می‌آیند.

```
zigma: 205  
best zigma is 0 and worst zigma is: 29929930
```

در این قسمت نیز خود کد نکته خاصی ندارد و متشکل از دستورات ساده و متداول بش است.

سوال ۳:

خط اول نکته خاصی ندارد، از خط ۴ تا ۷ نوع خاصی از حلقه `while` وجود دارد که خط به خط محتویات فایل را می‌خواند و به `file_text` اضافه می‌کند. سپس در `file_text` به یک آرایه داده می‌شود تا هر کلمه در یک خانه آرایه جای گیرد، سپس روی آرایه یک حلقه `for` زده می‌شود و مطابقت تک‌تک کلمات با دو کلمه مورد نظر بررسی می‌شود و به ازای هر بار مطابقت یک واحد به متغیر `output` اضافه می‌کند و نهایتاً `output` را چاپ می‌کند.

سوال ۴:

در ۷ خط اول ابتدای وارد دایرکتوری Documents/scrap می شویم. (ابتدا آن را می سازیم.) سپس در خطوط ۹ تا ۱۹ عمل scrapping را با استفاده از ابزار wget انجام می دهیم که در ابتدا نام ابزار و در انتها آدرس وبگاه آمده است. توضیح گزینه های مورد استفاده در این ابزار:

- recursive-- : به معنای این است که پس از ورودی به یک صفحه به صورت بازگشتی باقی صفحات متصل به آن را پیمایش می کنیم.
- wait-- : فاصله زمانی بین هردو درخواستی که ارسال می شود، در صورت ارسال بدون توقف درخواست های ممکن است سایت ما را بلاک کند، پس با استفاده از این گزینه بین درخواست ها فاصله ای در نظر می گیریم.
- user-agent-- : با استفاده از این گزینه می توان هدر User-agent را در HTTP request که نشان دهنده ارسال کننده درخواست است را تغییر داد.
- level-- : عمق جستجوی را مشخص می کند.
- e robots-- : برخی سایت ها آدرس برخی صفحات را در فایل robots.txt قرار می دهند و از ربات ها می خواهند هنگام پیمایش صفحات سایت آن صفحات را پیمایش نکنند، با غیر فعال کردن این گزینه wget صفحاتی که سایت از ربات های خواسته آن ها را پیمایش نکنند را هم پیمایش می کند.
- regex-type-- : تعیین کننده expression type مورد استفاده در رجکس.
- reject-regex-- : در اینجا می توان یک expression خاص برای URL تعیین کرد تا اگر wget حين پیمایش به آن برخورد کرد صفحات مربوطه را پیمایش نکند.
- no-directories-- : با فعال شدن این گزینه فایل های دانلود شده بدون فولدر بندی همه در همان دایرکتوری که دستور اجرا شده قرار داده می شوند.
- html-extention-- : به فایل های دانلود شده بدون پسوند psوند html را اضافه می کند، دلیل اینکه چرا برخی فایل ها ممکن است این پسوند را نداشته باشند ولی محتویات آن ها html باشد در صفحه راهنمای wget آمده است.

سپس رو تمام فایل های html دانلود شده پیمایش می کنیم و با استفاده از pandoc آن ها را به فایل تکست تبدیل می کنیم. -f و -t به ترتیب خلاصه شده from و to هستند و فرمت فایل ورودی و خروجی را تعیین می کنند. -O هم تعیین می کند که خروجی به صورت فایل باشد و stdout نباشد، سپس آدرس فایل های ورودی و خروجی داده می شود.

سپس در خط ۳۰ با استفاده از ابزار cat محتویات تمام فایل های تکست ساخته شده را در یک فایل تکست به نام ALL.txt می ریزیم و سپس در خط ۳۱ ابتدا با استفاده از tr ابتدا تمام علائم نگارشی و فاصله ها را با اینتر جایگزین می کنیم، سپس آن را سورت می کنیم و سپس با استفاده از ابزار uniq تعداد تکرار هر کلمه یکتا را می شماریم. (-C به معنای count است.) در انتهای دستور نیز آدرس فایل خروجی تعیین می شود.

سوال ۵:

در خط ۷-A به معنای این است که همه فایل‌های اضافه‌نشده را به گیت اضافه کند و به معنای «همه» است.

در خطوط ۸ و ۵۰-m پیام کامیت را تعیین می‌کند.

در خط ۱۸-b تعیین می‌کند که قبل از چک‌اوت کردن برنج را بساز.

بقیه دستورات نکته خاصی ندارند و دستورات ساده و متداول گیت و بش هستند.