Using the regex pattern notation discussed in class, write a regular expression that could be used to match each of the following.

(a) Any sequence of letters (upper- or lowercase) that includes the letter Z (in uppercase).

Answer: `[a-zA-Z]*Z[a-zA-Z]*`

Discussion: The pattern `[a-zA-Z]` matches any letter in the ranges a-z or A-Z.

(b) Any eleven-digit telephone number written in the form `(xxx)xxx-xxxx`.

Answer: `\([0-9][0-9][0-9]\)[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]` or `\([0-9]{3}\)[0-9]{3}-[0-9]{4}`

Discussion: Remember that `(` is a meta-character and must be escaped. The second form uses the $\{n\}$ notation to indicate a specific count.

(c) Any eleven-digit telephone number either in the form `(xxx)xxx-xxxx` or `xxx-xxx-xxxx`.

Answer: `\([0-9]{3}\)[0-9]{3}-[0-9]{4}|[0-9]{3}-[0-9]{3}-[0-9]{4}` or `(\([0-9]{3}\)|[0-9]{3}-)[0-9]{3}-[0-9]{4}`

Discussion: The second form reflects the observation that the `xxx-xxxx` part of both formats is the same. Note the use of both `()` to group elements and `\(\)` for literal parens.

Give a search pattern and a replace pattern that could be used to convert seven-digit telephone numbers in the format `xxx-xxx-xxxx` to the format `(xxx)xxx-xxxx`.

Answer: The search pattern is `([0-9]{3})-([0-9]{3}-[0-9]{4})` and the replace pattern is `$1-$2`

Discussion: The entire string matching the search pattern is replaced by the replace pattern. Use `()` to denote groups of matched elements that should appear in the result, and use `$1`, `$2`, etc to refer to those matched groups.

Write a pattern that matches all strings in the language $L = \{a^n b a^n \mid n \geq 0\}$.

Answer: `(a*)b\1`

Discussion: `(a*)b(a*)` matches all strings containing exactly one b, but it doesn't require that the number of as before and after the b be the same. The backreference `\1`, which refers to what was matched by the first set of parens, makes it possible to specify that the substring following the b must be the same as whatever substring preceeded the b.