Re Replacing

***don't forget to submit you progress at gaming project checkpoints!!!

***the TAs strongly suggest you use http://pythex.org/ instead of regexr

Solution to our problem yesterday!

Getting computing ids but only if that's the whole thing, and works for beginning and end:

[a-z]{2,3}[0-9][a-z]{1,2}

Two scenarios: one, there is nothing at the end; two, there is something we don't want so use an or |

[a-z]{2,3}[0-9][a-z]{1,2}([^a-z]|$)

So it ends with "either something not a letter or the end of the string"

***I proposed another possible issue and we realized there are infinite possible issues, but if they're not present in this data set, it's not important--so always look at your data to decide what to fix

You need to put a parentheses around the options in your or statement, or it will make the first option everything up to the | and the second option everything after it

Otherwise, parentheses are used for grouping--I think I put in an earlier one that they're otherwise irrelevant, that's not true; they are crucial in or statements

Parentheses also determine what a **+ *** or **?** apply to, so watch them there, too

Example of replacing regex expressions with new function **regex_name.sub(r'the expression', the_string)**:

```
# take a string, look for matches, if we find the, replace them
import re

text = 'Special Agent Upsorn told Special Agent Stephanie a secret'
# we want to censor the names
# replace the name with 'REDACTED'

regex = re.compile(r'(Agent) (([A-Z][a-z]+ )+)')
# first + adds letters after the capital, second + adds names (e.g., Agent Mary
Beth)
# in two groups, the first is the word 'Agent' and the second is the name
for match in re.finditer(regex, text):
   print(match.group())
   # .group() is just the string, .start() and .end() give indices
   # prints Agent Upsorn and then on another line Agent Stephanie

string = regex.sub(r'REDACTED ', text)
print(string)
# prints 'Special REDACTED told Special REDACTED a secret'

string2 = regex.sub(r'\1 REDACTED ', text)
print(string2)
# prints 'Special Agent REDACTED told Special Agent REDACTED a secret
```

```
# \1 means the first group, so we are replacing the whole thing with the first
group, then a space, then REDACTED
# we can do this with any group, or use single characters as groups

# let's get just the first letter (need to change grouping)
regex = re.compile(r'(Agent )([A-Z])[a-z]+')
string3 = regex.sub(r'\1\2*****', text)
print(string3)
# prints 'Special Agent U***** told Special Agent S***** a secret'

# try seeing what else you can make it print by changing groups as at-home
practice

# note: if there are characters before the first paran, they are not part of any
group
```

Practice of the Day 20:

```
import urllib.request
import re

def count_IP_addresses(url):
    '''
    returns a list of unique IP addresses in the file
    '''
    stream = urllib.request.urlopen(url)
    regex = re.compile(r'[0-9]{1,3}\.([0-9]{1,3}\.){2}[0-9]{1,3}')
    lst = []
    for line in stream:
        decoded = line.decode('UTF-8').strip()
        for match in re.finditer(regex, decoded):
            lst.append(match.group())
    return lst

print(count_IP_addresses("http://cs1110.cs.virginia.edu/code/access.log"))
# this is a log of office hours help
# note: the thing this prints is really long and takes a while, but trust me, this
works
```