

:[13] In

```
import re

#Exercise 1: For a given sentence, demonstrate how to applyRegEx.

txt = "Regular expression is a sequence of character(s) mainly used to find and replace pat

x = re.search("^Regular.*file$", txt)

if x:
    print("YES! We have a match!")
else:
    print("No match")
```

!YES! We have a match

:[14] In

```
#Replacing words using regular expressions

import re
replacement_patterns = [
(r'won\t', 'will not'),
(r'can\t', 'cannot'),
(r'i\m', 'i am'),
(r'ain\t', 'is not'),
(r'(\w+)\ll', '\g<1> will'),
(r'(\w+)n\t', '\g<1> not'),
(r'(\w+)\ve', '\g<1> have'),
(r'(\w+)\s', '\g<1> is'),
(r'(\w+)\re', '\g<1> are'),
(r'(\w+)\d', '\g<1> would')]

class RegexReplacer(object):
    def __init__(self, patterns=replacement_patterns):
        self.patterns = [(re.compile(regex), repl) for (regex, repl)in patterns]
    def replace(self, text):
        s = text
        for (pattern, repl) in self.patterns:
            (s, count) = re.subn(pattern, repl, s)
        return s

replace = RegexReplacer()
```

:[8] In

```
#Exercise 2: Demonstrate how can use the source code above to replace the sentences below:

Sentence=''We'll see how to replace words using regular
expressions such doesn't, can't and so on''

print(replacer.replace(Sentence))
```

We will see how to replace words using regular  
expressions such does not, cannot and so on

:[9] In

```
#Dealing with repeating characters

class RepeatReplacer(object):
    def __init__(self):
        self.repeat_regexp = re.compile(r'(\w*)(\w)\2(\w*)')
        self.repl = r'\1\2\3'
    def replace(self, word):
        repl_word = self.repeat_regexp.sub(self.repl, word)
        if repl_word != word:
            return self.replace(repl_word)
        else:
            return repl_word

repeat = RepeatReplacer()
```

:[12] In

```
#Exercise 3: Get a sentence from the user. If the sentence contains any repeating characters
Sentence = "We likkkkkke python"
print(repeat.replace(Sentence))
```

We like python

:[ ] In