Pattern matching - Exercises

Exercise 1
Check if the SRC_HUMAN sequence above has the following P-site: 'R(.)[ST][^P]'.
What amino acid is in the second position of the occurrence?

Exercise 2
Extract the title and the abstract text from a PubMed HTML page
If you go to a PubMed abstract web page (e.g. http://www.ncbi.nlm.nih.gov/pubmed/18235848) and click View -> View source, in the browser menu, the corresponding HTML source code will appear. Spend few minutes exploring this page. You will see that the title of the paper is enclosed between the following tags: <h1> and </h1> whereas the text of the abstract is enclosed between <h3>Abstract</h3> and </p>. These details are relevant for the selective extraction of the title and the abstract from a PubMed HTML abstract page.

Define and use a function that reads the HTML source page of PMID 18235848 and writes to a new file the corresponding title and the abstract text.

**Hint:** Here is the simple code to fetch the HTML source of a web page, read it as a single string and store it in a variable (html):

```python
import urllib2
PMID = 18235848
handler = urllib2.urlopen(url)
html = handler.read()
```

Exercise 3
Use the function defined in Exercise 2 to extract title and abstract for the following PMIDs: '18235848','22607149','22405002','21630672'.

Exercise 4
Detect a specific word or a set of words in a scientific abstract
Search the words [Ss]chistosoma in the scientific abstracts of Exercise 2 and report the result in the form:
PMID 18235848 occurrence start end
...
Solutions

Exercise 1

Exercise 5
Check if the SRC_HUMAN sequence above has the following P-site: 'R(\.)[ST][^P]'.
What amino acid is in the second position of the occurrence?

```python
import re
F = open('SRC_HUMAN.fasta')
seq = ''
for line in F:
    if line[0] != '>':
        seq = seq + line.strip()
S = 'R(\.)[ST][^P]'
regexp = re.compile(S)
iter = regexp.finditer(seq)
for s in iter:
    print s.group(0), s.group(1), s.span()
```

Exercise 2

*Extract the title and the abstract text from a PubMed HTML page*
Define and use a function that reads the HTML source page of PMID 18235848 and writes to a new file the corresponding title and the abstract text.

```python
import urllib2, re
def fetch_pmid(PMID):
    handler = urllib2.urlopen(url)
    html = handler.read()
    title_regexp = re.compile('<h1>.{5,400}</h1>')
    title_text = title_regexp.search(html)
    abstract_regexp = re.compile('<h3>Abstract</h3><p>.{20,3000}</p>')
    abstract_text = abstract_regexp.search(html)
    return title_text.group(), abstract_text.group()

title, abstract = fetch_pmid('18235848')
print title, abstract
```

Exercise 3

Use the function defined in Exercise 2 to extract title and abstract for the following PMIDs:
'18235848','22607149','22405002','21630672'.

```python
import Exercise_2
pmids = ['18235848','22607149','22405002','21630672']
for pmid in pmids:
    title, abstract = Exercise_2.fetch_pmid(pmid)
    print title, abstract
```
Exercise 4

Detect a specific word or a set of words in a scientific abstract

Search the words [Ss]chistosoma in the scientific abstracts of Exercise 3 and report the result in the form:

PMID  18235848  occurrence start end

```python
import urllib2, re
import Exercise_2

pmids = ['18235848','22607149','22405002','21630672']

for pmid in pmids:
    title, abstract = Exercise_2.fetch_pmid(pmid)
    word_regexp = re.compile('[Ss]chistosoma')
    word = word_regexp.search(abstract)
    if word:
        print title
        print word.group(), word.start(), word.end()
    else:
        print title
        print "Not found"

#This is if you want to find ALL occurrences
for pmid in pmids:
    title, abstract = fetch_pmid(pmid)
    word_regexp = re.compile('SmTGR')
    all_words = word_regexp.findall(abstract)
    words = word_regexp.finditer(abstract)
    if all_words:
        print pmid,
        for word in words:
            print word.group(), word.start(), word.end()
    else:
        print pmid
        print "Not found"
```