Regular Expression Practice Exercises

Instructions

These three tasks are supplementary exercises allowing you to practice regular expressions in real-life cases. Attempt to construct JavaCC RegExps to meet the following three tasks, and discuss the solutions on Cityspace. I will be monitoring and providing input from time to time, but I expect you to support and discuss with each other using the boards. Group work is permitted, and in fact encouraged!

As an entirely extra-curricular task,(Read: this will not be on the exam!) you might wish to investigate perl-compatible regular expressions (useful if you are developing websites using Java, JSP or PHP) - http://en.wikipedia.org/wiki/PCRE

Task 1: UK Postcodes

UK postcodes have an extremely predictable syntax. All postcodes are in upper case only.

1. London Postcodes (excluding EC- and WC-), consist of the prefix **N**, **NW**, **W**, **SW**, **SE**, **E** followed by one or two numbers, a space, a number and a letter: **N1 3LS**, **NW1 8TQ**, **E17 2DF**, **SE24 5TH**, **SW7 2SE**
2. National Postcodes: One or two letters, followed by one or two numbers, followed by a space, a number and two letters. e.g. **SL6 9EF**, **AB10 3ER**, **B11 6TH**
3. Inner London postcodes - find out the format of the inner London (**EC**- and **WC**-) postcodes: e.g. **EC1V 0HB**

Construct regular expressions to match each of these three categories of postcode.

Task 2: UK Phone Numbers

Each expression for a phone number should be viewed as an addition to the previous expressions constructed.

1. National phone numbers (with no spaces or bracketing): 11 digits in length, always starting with a **0**. e.g. **02082319999**, **01628629999**, **01184349999**.
2. National phone numbers (with spacing): 11 digits in length, area code 3-5 numbers in length, digits optionally grouped in 3s or 4s (not 5s). e.g. **020 82319999**, **020 8231 9999**, **0118 4349999**, **0118 434 9999**, **01628 629999**, **01628 629 999**.
3. Any of number 2, with the area code surrounded with parenthesis. e.g. **(020) 82319999**, **(0118) 434 9999**, **(01628) 629999**. Optinally, there can be no space between the parenthesised area code and the number. e.g. **(020) 82319999**.
4. International dialling to the UK. Numbers are 10 digits in length (leading 0 is stripped out), and begins with the string **+44**.

Task 3: Email addresses

Email addresses are Complicated. They are defined in RFC 2822 (http://tools.ietf.org/html/rfc2822#section-3.4). Some simplified patterns will match the vast majority of addresses, and are worth considering as an exercise. Implement each of these seperately, and think about the limitations of each.

1. Very simple address matching; `a@b`. Any string of letters or numbers, followed by an @ symbol, followed by any string of letters/numbers.
2. Matching `user@host.domain` style addresses. Same as number 1, except that the right hand side must contain a dot.
3. Matching allowable top level domains only (e.g. com, net, org, biz, info, coop, uk, gov). Choose a couple of indicative country top level domains only, rather than listing them all!
4. Full RFC2822 compliance, or a close approximation.