CS 5/49995 – ST: Introduction to Bioinformatics

Practice Problems: Regular expressions and pattern matching

Solutions are included on the following pages

1) Write a PERL regular expression that would match *only* the strings: "cat", "at", and "t".

2) Write a PERL regular expression to recognize *any string* that contains the substring "bob".

3) Write a PERL regular expression that would match the strings: "cat", "caat", "caaat", "caa…aat", etc. (strings that start with c, followed by one or more a's, ending with a t).

4) Write a PERL regular expression that matches the strings: "dog", "Dog", "dOg", "DOG", "dOG", etc. (That is, "dog" written in any combination of uppercase or lowercase letters.)

5) Write a PERL regular expression that matches any positive number (with or without a decimal point). *Hint #1: if there is a decimal point, there must be at least one digit following the decimal point. Hint #2: Since the dot "." matches any character, you must use \. to match a decimal point.*

6) Write a PERL regular expression to match any integer that doesn't end in 9.

7) Write a PERL regular expression to match any line with exactly two words (or numbers) separated by any amount of whitespace (spaces or tabs). There may or may not be whitespace at the beginning or end of the line.

8) Given the regular expression: c(.*)gg(\d*)
and the string: c my dog run gg234and me

What will be the value of $1?

What will be the value of $2?

9) Give the regular expression: c[abc]+\s+k[aeiou]*
list five strings that would match this regular expression. List two strings that would *not* match this regular expression.

10) Write a Perl regular expression that would match any word that has only two vowels: one e and one a.

11) Write a Perl regular expression that would match any word that starts and ends with the same two letters.

Solutions

1) Write a PERL regular expression that would match the strings: "cat", "at", and "t".

   `/^c?a?t$/`

2) Write a PERL regular expression to recognize any string that contains the substring "bob".

   `/bob/`

3) Write a PERL regular expression that would match the strings: "cat", "caat", "caaat", "caa…aat", etc. (strings that start with c, followed by one or more a's, ending with a t).

   `/^ca+t$/`

4) Write a PERL regular expression that matches the strings: "dog", "Dog", "dOg", "DOG", "dOG", etc. (That is, "dog" written in any combination of uppercase or lowercase letters.)

   /^[dD][oO][Gg]$/

5) Write a PERL regular expression that matches any positive number (with or without a decimal point). *Hint: if there is a decimal point, there must be at least one digit following the decimal point. Hint #2: Since the dot "." matches any character, you must use* `\.` *to match a decimal point.*

   /^\d+(\.\d+)?$/

6) Write a PERL regular expression to match any integer that doesn't end in 9.

   /^\d*[^9]$/

7) Write a PERL regular expression to match any line with exactly two words (or numbers) separated by any amount of whitespace (spaces or tabs). There may or may not be whitespace at the beginning or end of the line.

   ^\s*\w+\s+\w+\s*$

8) Given the regular expression: c(.*)gg(\d*)
and the string: c my dog run gg234and me

What will be the value of $1? `‘ my dog run ‘`

What will be the value of $2? `‘234‘`

9) Give the regular expression: c[abc]+\s+k[aeiou]*
list five strings that would match this regular expression. List two strings that would *not* match this regular expression.

Matching strings:
```
cbba  keai
cbc kiu
cc     k
```

Strings that do not match:
```
bob
ted
```

10) Since we don't know if the e or the a will come first, we have to do this in two parts and then use | as an "or" operator to combine the parts:

/(^[^aeiou]*a[^aeiou]*e[^aeiou]*$|^[^aeiou]*e[^aeiou]*a[^aeiou]*$)/

11) /^(..).*\1$/