1. Which of the following matches regex /abababa/?
   1) abababa
   2) aaba
   3) aabbaa
   4) aba
   5) aabababa (However the first "a" in the String will not be matched by the regex "abababa" the rest will.)

2. Which of the following matches regex /ab+c?/?
   1) abc
   2) ac
   3) abbb
   4) bbc

3. Which of the following matches regex /a.[bc]+/?
   1) abc
   2) abbbbbbbb
   3) azc
   4) abcbcbcbc
   5) ac
   6) asccbbbbcbcccc

4. Which of the following matches regex /abc|xyz/?
   1) abc
   2) xyz
   3) abc|xyz

5. Which of the following matches regex /[a-z]+[\.\?!]/?
   1) battle!
   2) Hot
   3) green
   4) swamping.
   5) jump up. --- ("However, this whole String isn't highlighted only the "up." As "jump" isn't followed by any special character specified in the regex").
   6) undulate?
   7) is.? ---("This sentence is highlighted, however the "?" in the String would not be. This occurs due to the regex looking for any word followed by a specified symbol").

6. Which of the following matches regex /[a-zA-Z]*[^,]=/?
   1) Butt=
   2) BotHEr,=
   3) Ample
   4) FIdDlE7h=
   5) Brittle =
   6) Other.=

7. Which of the following matches regex /[a-z][\.\?!]\s+[A-Z]/?
   (\s matches any space character)
   1) A. B
   2) c! d
   3) e f
   4) g. H
   5) i? J
   6) k L

8. Which of the following matches regex /(very)+(fat)?(tall|ugly) man/?
   1) very fat man
   2) fat tall man
   3) very very fat ugly man
   4) very very very tall man

9. Which of the following matches regex /<[^>]+>/?
   1) <an xml tag>
   2) <opentag> <closetag>
   3) </closetag>
   4) <>
   5) <with attribute="77">

10. Write a regex to identify dates of the form dd/mm/yyyy.
    I expect dd to range from 01 to 31, and mm to range from 01 to 12, and I expect yyyy to range from 0001 to 9999 and in particular to *not* be 0000 (the Gregorian calendarpredates this; see Year 0 and the invention of 0).
    However, I do not expect you to cross-reference mm against dd or to restrict yyyy, so that e.g. 31/02/0231 is fine.
    Do **not** use backreferences or negative lookahead (so if your answer contains ?!, then it's not admissible for this question).

(0[1-9]|[12][0-9]|3[01])[/](0[1-9]|1[012])[/](([0-9][0-9][0-9][1-9])|([1-9][0-9][0-9][0-9])|([0-9][1-9][0-9][0-9])|([0-9][0-9][1-9][0-9]))

11. Write a regex to identify dates of the form dd/mm/yyyy or dd.mm.yyyy, but *not* using mixed separators such as dd/mm.yyyy.
    You may use backreferences, negative lookahead, and other fancy tricks, if convenient.

((([(0[1-9]|[12][0-9]|3[01])[/](0[1-9]|1[012])[/]((?!0000)\d{4})|(0[1-9]|[12][0-9]|3[01])[.](0[1-9]|1[012])[.]((?!0000)\d{4})))

12. *(Unmarked)* Explain whether a regex can identify correct bracketing, as in ((a)) but not (((a).

It can, if you look at programming languages such as SML the program will tell you there is syntactical errors within the program you are trying to compile. This is based upon recursion where once you open a bracket it will check each opening and closing brackets to see if they have been correctly closed. However, this can also give way to mild errors but this is based on the programmer.

13. *(Unmarked)* Explain the relevance of the regex Whiske?y

It allows for an 0 or 1 optional character that the special character follows. For example in the regex "Whiske?y", the "e" in the String is optional so therefore you could match the regex with either "Whiskey" or "Whisky".

14. Write a regex for the set of even numbers (base 10; so the alphabet is [0-9]). Note that 0 is an even number and 00 and 02 are not even numbers. Check your regex accepts 100 and 10012.

^([0]|\d+[24680]$)