PATTERN MATCHING

1) What is pattern matching ? Explain

Ans:

Pattern matching is a process of checking whether a specific sequence of characters/token/data exists among given data. It is also used to find and replace a matching pattern in a text or code with another text/code.

For ex, x* matches any number of x characters, [0-9]* matches any number of digits, & * matches any number of anything.

In Linux, wildcards allow you to specify succinctly a pattern that matches a set of filenames for ex, *.pdf to get a list of all the PDF files. It also often referred to as glob patterns.

Pattern Match can be used in identification as well as in pre-classification processing, Page processing, or Storage processing. When defining a pattern, we should only use the expressions defined under Regular Expression Reference. Wildcards are not supported for this process.

2) What are the 2 main characters used for matching Pattern and How correlation pattern matching works?

Ans: Pattern matching employs wildcard characters to match different combinations of characters. The LIKE keyword indicates that the following character string is a matching pattern.

Template matching is a basic and simple pattern matching technique in digital signal processing, in particular in digital image processing. The correlation approach uses correlation coefficient as a measure of similarity b/w the sequence (template) for each location (x,y) in target image.

3) Write Table of Metacharacters in Linux?

Ans:

1) ^ (caret) match beginning of line.
Anchors match

2) $ (dollar sign) match end of line.
Anchors match

3) $ (dollar sign) match end of line

3) . (dot) match any character. Because command line globbing uses ? instead.

4) * (star) matches zero or more of preceding characters. Because, command line uses * as in .* .

5) [] (square Braces) set of characters inside Braces, match any one of.

6) [^] (caret at first character inside braces), matches any character except those inside braces.

7) [a-z] (use of dash inside Braces) match a range. If - is to be matched, must be first character, to avoid misinterpretation as range operator.

8) () {parenthesis, must be escaped with Backslash), save match for later use with \n, where n is a number.

9) {m}, {m,} and {m,n} [Braces, which must be escaped with a Backslash), matched m, more than m, or between (m and n).

10) & (ampersand) expands to the matched string, used in sed.