Day 4.

1) What Pattern Matching ? Explain in detail.

Shell patterns are used in a number of contents. In a pattern, most characters match themselves and only themselves. The word hello is a prefectly valid pattern it matches the word hello and nothing else. A pattern that matches only part of a string is not considered to have matched that string. The word hello does not match the text hello, world. For a pattern to match a string, two things must be true:

Every character in the pattern must match the string.
Every character in the string must match the pattern.

Now, if this were all there where to patterns, a would be another way describing string comparision an the rest of this chapter would consists of filler text like 'a "consists of sequences of nonblank characters seperated by blanks" or possibly some wonderful Cookies recipes. Sadly this is not so. Instead, there are some characters in a pattern that have special meaning and can match something other than themselves. Characters that have special meaning and can match something other than themselves. Charaters that have special meaning in a pattern are called wildcard or metacharacters. Some users prefer to restrict the term wildcard to refer only to the special character that can match anything. In talking about patterns, I prefer to call them all wildcards to avoid confusion with character that have special meaning to the shell. Wildcards make these too simple rules much more complicated. a single character in a pattern could match a very long string or a group of characters in the

pattern might match only the one character or even none at all. What matters is that there are no mismatches and nothing left over of the string after the match

The most commond wildcards are the question mark(?) which matches any character, and asterisk(*) which matches anything at all, even an empty string.

2) What are wildcard characters? How can we use it?

A wildcard is a symbol that takes the place of an unknown character or set of characters Commonly used wild cards are the asterisk(*) and the question mark(?). Depending on the software or the search engine you are using, other wildcard characters may be defined. When you are searching for files in Unix, DOS or Windows or on the web, you can simply you search by using a wildcard. Wildcards may also simplify commands issued from the command line in UNIX or DOS.

The asterisk(*)

The asterisk represents any number of unknown characters. Use it when searching for documents or files for which you have only partial names.

Ex :-

Create 3 files name file1 file2 file3. To list out all the files ~~written~~ above created. Just enter file*

O/p → file1 file2 file3 all of them will be listed.

The question mark (?)

The question mark represents only one unknown character. Use it when you have a list of files with very similar names, or when you are unsure of a few characters.

Create 5 files named take1.txt taken.txt take2.txt take22.txt take33.txt take44.txt

If you enter take?.txt

Op take1.txt
take2.txt
taken.txt

will be displayed after take only one character will be matched and displayed.

take??.txt

take22.txt
take33.txt
take44.txt

will be displayed as ?? 2 characters are being matched and displayed.

Combining (*) and ?

You can use the asterisk (*) and the question mark (?) anywhere in a search, and you can also use them together. For example If you want to find all the files that starts with home followed by one or two characters and ending with the

any extension enter home??.* as you search term. Your search may or might return home45.bak or home4-txt but not homeloan-doc
as it more than 2 characters.

3) Assumption file names as abc1, abc2, abc3, abc4 ... abc8 - command for files using in range in the char class remove the files using char class and files range.

Assume you have parent1, parent2, parent3 as directories.
Copy to the parent dir thoes files having extension with at least 1 character. At least one char before the . but not having 1238 as last char.

mkdir parent1 → cd parent1
mkdir parent2 → cd parent2.
~~mkdir parent3~~
touch abc1
touch abc2
touch abc3
touch abc4
touch abc5
touch abc6
touch abc7
touch abc8

cp ?*.*[!1238]..