Worksheet 3: Regex

Updated: 25th February, 2020

Note: You are encouraged to keep notes on your observations and answers to the worksheet exercises and lecture examples.

Assistance is only given when students can demonstrate their own attempt of resolving any problems.

Make sure you include things that went wrong. Your notes will serve as evidence of attempt, and often you can learn a lot analysing things that did not work.

Your notes should include answers to the following questions:

- What programs or websites did you use in the practical today?
- What do you need to remember about your practical work today?
- How does your program work?

You should consult the lecture notes whilst attempting these exercises.

1. `grep`

(a) Work out what the following commands do, then test your understanding by trying them. You will have to contrive a text file for this (or copy and paste the text at the end of this worksheet).

```
1 grep "[mM].*[Kk]" file.txt
2 grep "\<[mM].* [Kk]\>" file.txt
3 grep "[mM].\{2\}[kK]" file.txt
4 grep -v "[mM].\{2\}[Kk]" file.txt
```

(b) Use **`grep`** to match lines containing:

(i) `Jefferies jeffery jeffeys`

(ii) `hitchen Hitchin hitching`

(iii) `Heard herd Hird`

(iv) `dix dicks Dickson dixon`

(v) `Mcgee mcghee magee`

(c) Use command substitution with **`grep`** to output the lines of the persons from `file.txt` who have their birthday today. Optional: `man cut` to find out how to output only the name.

(d) Find the occurrences of three consecutive and identical word characters (like aaa or bbb).

(e) Write a command to output the lines that are 50 characters in length or longer.

2. vi

(a) Work out what the following commands do, then test your understanding by trying them. You will have to contrive a text file for this (or copy paste the text at the end of this worksheet)

```
1 s/ */ /g
2 s/^/ /
3 %s/^[0-9][0-9]* //
4 s/b[aeio]g/bug/g
5 s/t\([aou]\)g/h\lt/g
```

(b) Write a command to delete all the c++ style comments in a file

(c) Try the "Whole word only" example in the lecture notes

3. sed

(a) Write a command to output lines that are less than 50 characters in length.

(b) Find out the occurrences of three consecutive and identical word characters (like aaa or bbb).

(c) What is wrong with this command? How would you correct it?

```
[user@pc]$ sed 's/compute/calculate/g s/computer/host/g' file.txt
```

(d) Write a command sequence to find out the number of occurrences of the word "encryption" in a file. (Note: A word may occur more than once in a line.)

Note: For interest. The following code adds line number to the start of each line of `myprog.c`

```
[user@pc]$ sed = myprog.c | sed 'N;s/\n/\t/' > myprog.c
```

4. c

Read the tutorial: GNU Regex. The following function returns true when `pattern` matches `string`

```c
int match(const char *string, char *pattern)
{
   int match = FALSE;
   regex_t re;
   if (regcomp(&re, pattern, REG_EXTENDED|REG_NOSUB) == 0)
   {
       if (regexec(&re, string, (size_t)0, NULL, 0) == 0)
       {
           match = TRUE;
           regfree(&re);
       }
   }
   return match;
}
```

(a) Write a complete c program to test this function. Your program should read both the string to match and the pattern from user input.

> **Note:** You should use **`fgets()`** to read a line of text from **`stdin`**. **`fgets()`** is safer than **`gets()`** and **`scanf()`** doesn't read spaces, etc.

(b) Modify this function to report **where** a sub expression matches.

> **Note:** Use the 'banana' example in the reference above.

```
// file.txt
// if your regex doesn't work, make sure the lecturer (mark) has included
// the appropriate text here

Steve Jefferies:923-7366:95 Latham Lane, Easton, WA 6032:11/12/56
Betty encryption:836-8357:635 encryption Lane, Hollywood, WA 1464:6/23/23
Igor Chevsky:375-8395:3567 encryption Place, Caldwell, WA 3875:6/18/68

Norma Korder:857-2735:74 Pine Street, Dearborn, WA 3874:3/28/45
Jennifer Cowan:834-2348:583 Laurel Ave., Kingsville, NT 3745:10/1/35
Jon DeLoach:253-3122:123 Park St., San Jose, CA 4086:7/25/53
Karen Evich:758-2857:23 Edgecliff Place, Lincoln, NSW 2743:7/25/53
Karen Evich:758-2867:23 Edgecliff Place, Lincoln, NSW 2743:11/3/35
Karen Evich:758-2867:23 Edgecliff Place, Lincoln, NSW 2743:11/3/35

bag
beg
big
bog

Fred Fardbarkle:843-1385:20 Parak Lane, Duluth, MN 3850:4/12/23
magee Fardbarkle:843-1385:20 Parak Lane, Duluth, MN 3850:4/12/23
Mark Gortz:832-5728:3465 encryption Street, Peabody, MA 4756:10/2/65
Jeffery Gutierrez:365-1284:454 Easy Street, dixon, WA 5732:2/28/53
mcghee Hardy:259-5395:235 CarltonLane, Joliet, WA 3858:8/12/20
James Ikeda:938-8376:23445 Aster Ave., Allentown, NT 3745:08/26/38

tag
tog
tug

Barbara Kertz:573-8326:832 Jeffeys Drive, Gary, WA 3756:12/1/46
Lesley Kirstin:456-1234:4 Harvard Square, Dickson, WA 2133:08/28/62
William Kopf:836-2837:6937 hitching Road, Milton, WA 3756:9/21/46
Sir Lancelot:835-8257:474 Camelot Boulevard, Bath, WA 8356:5/13/69
Jesse Neal:233-8971:45 Rose Terrace, San Francisco, WA 2303
Zippy Pinhead:823-8319:2356 Bizarro Ave., Farmount, WA 4357
Arthur Putie:835-8745:23 Wimp Lane, Kensington, QLD 8758:8/31/69

//this is another comment
compute the result of PI * r * r on the computer moses
you should attempt encryption using encryption services

Popeye Sailor:454-3322:945 hitchen Street, Anywhere, QLD 9358:3/19/35
Jose Hird:898-8357:38 Heard Way, Abilene, QLD 9673
Tommy Mcgee:724-0140:1222 herd Court, dicks, QLD 4087:5/19/66
Yukio Takeshida:827-1095:13 dix Lane, Ashville, WA 3556
Vinh Tranh:910-7449:8235 Hitchin Street, Wilmington, WA 9085:9/23/63
```

End of Worksheet