Week 01 Laboratory Sample Solutions

Objectives

- Understanding regular expressions
- Understanding use of UNIX filters (grep)

Preparation

Before the lab you should re-read the relevant lecture slides and their accompanying examples.

Getting Started

Set up for the lab by creating a new directory called `lab01` and changing to this directory.

```
$ mkdir lab01
$ cd lab01
```

There are some provided files for this lab which you can fetch with this command:

```
$ 2041 fetch lab01
```

If you're not working at CSE, you can download the provided files as a zip file or a tar file.

EXERCISE:
grep-ing a Dictionary

You have been given a file named `dictionary_answers.txt` .
Which you must use to enter the answers for this exercise.

The autotest scripts depend on the format of `dictionary_answers.txt` .
So just add your answers where indicated but don't otherwise change the file.

```
    # Open a text editor (gedit) in the background (&) and not owned by the current terminal (disown)
    $ gedit dictionary_answers.txt & disown
    # Or use any other text editor of your choosing
```

On most Unix systems you will find one or more dictionaries containing many thousands of words:
Typically in the directory `/usr/share/dict/`

```
    $ ls -1 /usr/share/dict/
    README.select-wordlist
    british-english
    cracklib-small
    words -> /etc/dictionaries-common/words -> /usr/share/dict/british-english
```

We've created an example dictionary named `dictionary.txt` for this lab exercise.

1. Write a `grep -E` command that prints the words which contain the characters "lmn" consecutively.

HINT:

It should print:

```
Selmner
Selmner's
almner
almners
calmness
calmness's
calmnesses
```

ANSWER:

Sample answer:

```
$ grep -E 'lmn' dictionary.txt
```

The COMP2041 class account contains a script named **autotest** that automatically runs tests on your lab exercises.

Once you have entered you answer you can check it like this:

```
$ 2041 autotest dictionary Q1
Test Q1 (dictionary Q1) - passed
1 tests passed 0 tests failed
```

2. Write a `grep -E` command that prints the words which contain any four consecutive vowels.

HINT:

It should print:

```
Aiea
Aiea's
Araguaia
Araguaia's
Douai
Douai's
Graeae
Graiae
```

ANSWER:

Sample answer:

```
$ grep -E -i '[aeiou]{4}' dictionary.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest dictionary Q2
Test Q2 (dictionary Q2) - passed
1 tests passed 0 tests failed
```

3. Write a `grep -E` command that prints the words which contain all 5 vowels "aeiou" in that order.

The words may contain more than 5 vowels but they must contain "aeiou" in that order.

HINT:

It should print:

```
abstemious
abstemiously
abstemiousness
abstemiousness's
abstemiousnesses
abstentious
adenocarcinomatous
adventitious
```

ANSWER:

Sample answer:

```
            $ grep -E -i 'a.*e.*i.*o.*u' dictionary.txt
```

Once you have entered you answer you can check it like this:

```
            $ 2041 autotest dictionary Q3
            Test Q3 (dictionary Q3) - passed
            1 tests passed 0 tests failed
```

4. Write a `grep -E` command that prints the words which contain the vowels "aeiou", in that order, and no other vowels.

HINT:

It should print:

```
abstemious
abstemiously
abstentious
arsenious
caesious
facetious
facetiously
```

ANSWER:

Sample answer:

```
            $ grep -E -i '^[^aeiou]*a[^aeiou]*e[^aeiou]*i[^aeiou]*o[^aeiou]*u[^aeiou]*$'
dictionary.txt
```

Once you have entered you answer you can check it like this:

```
            $ 2041 autotest dictionary Q4
            Test Q4 (dictionary Q4) - passed
            1 tests passed 0 tests failed
```

When you think your program is working, you can use `autotest` to run some simple automated tests:

```
$ 2041 autotest dictionary
```

When you are finished working on this exercise, you must submit your work by running `give` :

```
$ give cs2041 lab01_dictionary dictionary_answers.txt
```

before **Tuesday 14 June 12:00** to obtain the marks for this lab exercise.

SOLUTION:

Sample solution for `dictionary_answers.txt`

```
This file is automarked.

Do not add extra lines to this file, just add your answers.

For example if your answer to Q0 is: "grep -E Andrew words.txt"
Change the line that starts with
    "Q0 answer:"
to
    "Q0 answer: grep -E Andrew words.txt"
----------------------------------------------------------------------------------------------------
--------------------



1) Write an grep -E command that prints the words which contain the characters "lmn" consecutively.
Q1 answer: grep -E 'lmn' dictionary.txt


2) Write an grep -E command that prints the words which contain any four consecutive vowels.
Q2 answer: grep -E -i '[aeiou]{4}' dictionary.txt


3) Write an grep -E command that prints the words which contain all 5 vowels "aeiou" in that order.
Q3 answer: grep -E -i 'a.*e.*i.*o.*u' dictionary.txt


4) Write an grep -E command that prints the words which contain the vowels "aeiou", in that order,
and no other vowels.
Q4 answer: grep -E -i '^[^aeiou]*a[^aeiou]*e[^aeiou]*i[^aeiou]*o[^aeiou]*u[^aeiou]*$' dictionary.txt
```

EXERCISE:

grep-ing Federal Parliament

You have been given a file named `parliament_answers.txt`.
Which you must use to enter the answers for this exercise.

The autotest scripts depend on the format of `dictionary_answers.txt`.
So just add your answers where indicated but don't otherwise change the file.

```
# Open a text editor (gedit) in the background (&) and not owned by the current terminal (disown)
$ gedit parliament_answers.txt & disown
# Or use any other text editor of your choosing
```

In this exercise you will analyze a file named `parliament.txt` containing a list of the members of the Australian House of Representatives (MPs).

NOTE:

As we have just had an election the information in the file `parliament.txt` might not be up to date.

1. Write a `grep -E` command that will print all the lines in the file where the electorate begins with 'W'.

HINT:

It should print:

```
Hon Scott Buchholz: Member for Wright, Queensland
Hon Tony Burke: Member for Watson, New South Wales
Mr Stephen Jones: Member for Whitlam, New South Wales
Mr Peter Khalil: Member for Wills, Victoria
Mr Llew O'Brien: Member for Wide Bay, Queensland
Mr Dave Sharma: Member for Wentworth, New South Wales
Ms Anne Stanley: Member for Werriwa, New South Wales
Ms Zali Steggall OAM: Member for Warringah, New South Wales
Hon Dan Tehan: Member for Wannon, Victoria
```

ANSWER:

Sample answer:

```
$ grep -E 'Member for W' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q1
Test Q1 (parliament Q1) - passed
1 tests passed 0 tests failed
```

2. Write a `grep -E` command that will print all the lines in the file where the MP's given name (first name) is "Andrew".

HINT:

It should print:

```
Hon Andrew Gee: Member for Calare, New South Wales
Mr Andrew Giles: Member for Scullin, Victoria
Hon Andrew Hastie: Member for Canning, Western Australia
Hon Dr Andrew Leigh: Member for Fenner, Australian Capital Territory
Hon Andrew Wallace: Member for Fisher, Queensland
Mr Andrew Wilkie: Member for Clark, Tasmania
```

ANSWER:

Sample answer:

```
$ grep -E '^((Mr|Mrs|Ms|Dr|Hon) )*Andrew .*:' parliament.txt
```

Note this more obvious answer will also match middle names

```
$ grep -E ' Andrew .*:' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q2
Test Q2 (parliament Q2) - passed
1 tests passed 0 tests failed
```

3. Write a `grep -E` command that will print all the lines in the file where the MP's surname (last name) ends in the letters 'll'.

HINT:

It should print:

```
Ms Angie Bell: Member for Moncrieff, Queensland
Mr Julian Hill: Member for Bruce, Victoria
Mr Brian Mitchell: Member for Lyons, Tasmania
Mr Rob Mitchell: Member for McEwen, Victoria
Ms Zali Steggall OAM: Member for Warringah, New South Wales
```

ANSWER:

Sample answer:

```
$ grep -E 'll( [A-Z]*)?:' parliament.txt
```

Note this more obvious answer does not handle the MP having an Order of Australia

```
$ grep -E 'll:' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q3
Test Q3 (parliament Q3) - passed
1 tests passed 0 tests failed
```

4. Write a `grep -E` command that will print all the lines in the file where the MP's surname (last name) **and** the electorate name ends in the letter 'y'.

HINT:

It should print:

```
Ms Peta Murphy: Member for Dunkley, Victoria
Mr Rowan Ramsey: Member for Grey, South Australia
```

ANSWER:

Sample answer:

```
$ grep -E 'y( [A-Z]*)?:.*y,' parliament.txt
```

Note this more obvious answer does not handle the MP having an Order of Australia

```
$ grep -E 'y:.*y,' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q4
Test Q4 (parliament Q4) - passed
1 tests passed 0 tests failed
```

5. Write a `grep -E` command that will print all the lines in the file where the MP's surname (last name) **or** the electorate name ends in the letter 'y'.

HINT:

It should print:

```
Dr Anne Aly: Member for Cowan, Western Australia
Hon Linda Burney: Member for Barton, New South Wales
Mr Pat Conroy: Member for Shortland, New South Wales
Mr Milton Dick: Member for Oxley, Queensland
Hon Ed Husic: Member for Chifley, New South Wales
Hon Bob Katter: Member for Kennedy, Queensland
Ms Ged Kearney: Member for Cooper, Victoria
Mr Craig Kelly: Member for Hughes, New South Wales
Hon Michelle Landry: Member for Capricornia, Queensland
Hon Sussan Ley: Member for Farrer, New South Wales
Mrs Melissa McIntosh: Member for Lindsay, New South Wales
Hon Ben Morton: Member for Tangney, Western Australia
Ms Peta Murphy: Member for Dunkley, Victoria
Mr Llew O'Brien: Member for Wide Bay, Queensland
Hon Tanya Plibersek: Member for Sydney, New South Wales
Mr Rowan Ramsey: Member for Grey, South Australia
Ms Michelle Rowland: Member for Greenway, New South Wales
Ms Anne Stanley: Member for Werriwa, New South Wales
Ms Anika Wells: Member for Lilley, Queensland
Mr Trent Zimmerman: Member for North Sydney, New South Wales
```

ANSWER:

Sample answer:

```
$ grep -E 'y( [A-Z]*)?:|y,' parliament.txt
```

Note this more obvious answer does not handle the MP having an Order of Australia

```
$ grep -E 'y[:,]' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q5
Test Q5 (parliament Q5) - passed
1 tests passed 0 tests failed
```

6. Write a `grep -E` command that will print all the lines in the file where there is any word in the MP's name or the electorate name that ends in "ng".

HINT:

It should print:

```
Hon Josh Frydenberg: Member for Kooyong, Victoria
Mr Luke Gosling OAM: Member for Solomon, Northern Territory
Hon Andrew Hastie: Member for Canning, Western Australia
Hon Catherine King: Member for Ballarat, Victoria
Ms Madeleine King: Member for Brand, Western Australia
Hon Bill Shorten: Member for Maribyrnong, Victoria
Mr Terry Young: Member for Longman, Queensland
```

ANSWER:

Sample answer:

```
$ grep -E 'ng[^a-z]' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q6
Test Q6 (parliament Q6) - passed
1 tests passed 0 tests failed
```

7. Write a `grep -E` command that will print all the lines in the file where the MP's surname (last name) both begins and ends with a vowel.

HINT:

It should print:

```
Hon Anthony Albanese: Member for Grayndler, New South Wales
```

ANSWER:

Sample answer:

```
$ grep -E '[AEIOU][^ ]*[aeiou]( [A-Z]*)?:' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q7
Test Q7 (parliament Q7) - passed
1 tests passed 0 tests failed
```

8. Write a `grep -E` command that will print all the lines in the file where the electorate name contains multiple words (separated by spaces or hyphens).

HINT:

It should print:

```
Hon Barnaby Joyce: Member for New England, New South Wales
Ms Kristy McBain: Member for Eden-Monaro, New South Wales
Mr Llew O'Brien: Member for Wide Bay, Queensland
Hon Matt Thistlethwaite: Member for Kingsford Smith, New South Wales
Hon Jason Wood: Member for La Trobe, Victoria
Mr Trent Zimmerman: Member for North Sydney, New South Wales
```

ANSWER:

Sample answer:

```
$ grep -E 'Member for [a-zA-Z]+[ -][a-zA-Z]' parliament.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest parliament Q8
Test Q8 (parliament Q8) - passed
1 tests passed 0 tests failed
```

When you think your program is working, you can use `autotest` to run some simple automated tests:

```
$ 2041 autotest parliament
```

When you are finished working on this exercise, you must submit your work by running `give` :

```
$ give cs2041 lab01_parliament parliament_answers.txt
```

before **Tuesday 14 June 12:00** to obtain the marks for this lab exercise.

SOLUTION:

Sample solution for `parliament_answers.txt`

```
This file is automarked.

Do not add extra lines to this file, just add your answers.

For example if your answer to Q0 is: "grep -E Andrew words.txt"
Change the line that starts with
    "Q0 answer:"
to
    "Q0 answer: grep -E Andrew words.txt"
----------------------------------------------------------------------------------------------------
--------------------


1) Write a grep -E command that will print all the lines in the file where the electorate begins
with 'W'.
Q1 answer: grep -E 'Member for W' parliament.txt


2) Write a grep -E command that will print all the lines in the file where the MP's first name is
"Andrew".
Q2 answer: grep -E '^((Mr|Mrs|Ms|Dr|Hon) )*Andrew .*:' parliament.txt


3) Write a grep -E command that will print all the lines in the file where the MP's surname (last
name) ends in the letters 'll'.
Q3 answer: grep -E 'll( [A-Z]*)?:' parliament.txt


4) Write a grep -E command that will print all the lines in the file where the MP's name and the
electorate ends in the letter 'y'.
Q4 answer: grep -E 'y( [A-Z]*)?:.*y,' parliament.txt


5) Write a grep -E command that will print all the lines in the file where the MP's name or the
electorate ends in the letter 'y'.
Q5 answer: grep -E 'y( [A-Z]*)?:|y,' parliament.txt


6) Write a grep -E command that will print all the lines in the file where there is any word in the
MP's name or the electorate name that ends in "ng".
Q6 answer: grep -E 'ng[^a-z]' parliament.txt


7) Write a grep -E command that will print all the lines in the file where the MP's surname (last
name) both begins and ends with a vowel.
Q7 answer: grep -E '[AEIOU][^ ]*[aeiou]( [A-Z]*)?:' parliament.txt


8) Write a grep -E command that will print all the lines in the file where the electorate name
contains multiple words (separated by spaces or hyphens).
Q8 answer: grep -E 'Member for [a-zA-Z]+[ -][a-zA-Z]' parliament.txt
```

EXERCISE:
Exploring Regular Expressions

You have been given a file named `ab_answers.txt` .
Which you must use to enter the answers for this exercise.

The autotest scripts depend on the format of `dictionary_answers.txt` .
So just add your answers where indicated but don't otherwise change the file.

```
# Open a text editor (gedit) in the background (&) and not owned by the current terminal (disown)
$ gedit ab_answers.txt & disown
# Or use any other text editor of your choosing
```

Use `grep -E` to test your answers to these questions.

We've provided a set of test cases in `input.txt`

1. Write a `grep -E` command that prints the lines in a file named `input.txt` containing at least one `A` and at least one `B`.

Matching	Not Matching
AB	A
BA	B
ABBA	AA
BANANA	Andrew
Andrew's favourite Band is not	George is Brilliant

ANSWER:

Sample answer:

```
$ grep -E 'A.*B|B.*A' input.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest ab Q1
Test Q1 (ab Q1) - passed
1 tests passed 0 tests failed
```

2. Write a `grep -E` command that prints the lines in a file named `input.txt` containing only the characters `A` and `B` such that all pairs of adjacent `A`'s occur before any pairs of adjacent `B`'s.

   In other words if there is pair of `B`'s on the line, there can not be a pair of `A`'s afterwards.

Matching	Not Matching
A	BBAA
ABBA	ABBAA
ABAABAABAABBBBABB	ABBABABABABAA
ABAAAAAAAAAABBA	ABBBAAA
ABABABABA	BBABABABABABABAA

ANSWER:

Sample answer:

```
$ grep -E '^(BA|A)*(BA|B)*$' input.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest ab Q2
Test Q2 (ab Q2) - passed
1 tests passed 0 tests failed
```

3. Write a `grep -E` command that prints the lines in a file named `input.txt` containing only the characters `A` and `B` such that the number of `A`'s is divisible by `4`.

Matching	Not Matching

AAAA	A
BABABABAB	AAAAA
AAAABBBBAAAA	ABABBBBBBBBBBBBBBBBAAA
BBBAABBBBBAABBBAAAA	AAAABBABBAAAA
B	BBBAABBABBBAABBBAAAA

ANSWER:

Sample answer:

```
$ grep -E '^B*(AB*AB*AB*AB*)*$' input.txt
```

Once you have entered you answer you can check it like this:

```
$ 2041 autotest ab Q3
Test Q3 (ab Q3) - passed
1 tests passed 0 tests failed
```

When you think your program is working, you can use `autotest` to run some simple automated tests:

```
$ 2041 autotest ab
```

When you are finished working on this exercise, you must submit your work by running `give` :

```
$ give cs2041 lab01_ab ab_answers.txt
```

before **Tuesday 14 June 12:00** to obtain the marks for this lab exercise.

SOLUTION:

Sample solution for `ab_answers.txt`

```
This file is automarked.

Do not add extra lines to this file, just add your answers.

For example if your answer to Q0 is: "grep -E Andrew words.txt"
Change the line that starts with
    "Q0 answer:"
to
    "Q0 answer: grep -E Andrew words.txt"
-----------------------------------------------------------------------------------------------------------------------



1) Write a grep -E command that prints the lines in a file named input.txt containing at least one A
and at least one B.
Q1 answer: grep -E 'A.*B|B.*A' input.txt


2) Write a grep -E command that prints the lines in a file named input.txt containing only the
characters A and B such that all pairs of adjacent A's occur before any pairs of adjacent B's.
Q2 answer: grep -E '^(BA|A)*(BA|B)*$' input.txt


3) Write a grep -E command that prints the lines in a file named input.txt containing only the
characters A and B such that the number of A's is divisible by 4.
Q3 answer: grep -E '^B*(AB*AB*AB*AB*)*$' input.txt
```

Submission

When you are finished each exercises make sure you submit your work by running `give`.

You can run `give` multiple times. Only your last submission will be marked.

Don't submit any exercises you haven't attempted.

If you are working at home, you may find it more convenient to upload your work via give's web interface.

Remember you have until **Week 3 Tuesday 12:00:00** to submit your work.

You cannot obtain marks by e-mailing your code to tutors or lecturers.

You check the files you have submitted here.

Automarking will be run by the lecturer several days after the submission deadline, using test cases different to those `autotest` runs for you. (Hint: do your own testing as well as running `autotest`.)

After automarking is run by the lecturer you can view your results here. The resulting mark will also be available via give's web interface.

Lab Marks

When all components of a lab are automarked you should be able to view the the marks via give's web interface or by running this command on a CSE machine:

```
$ 2041 classrun -sturec
```