

Perl Exercises

Lab 3

Log into Linux and create a file called **data.pl** with the following contents:

```
#!/usr/bin/perl

while (<DATA>)
{
    print if /data/; #This is the line to change.
}
__END__
This is the data this program will use.
As we are using the DATA filehandle, Perl looks to
the end of the script, represented by __END__, and
starts reading data from there, i.e., after __END__,
as if it was an input file.
This can be really handy when testing a script.
We will use it a lot.
```

Change the executable status of the file, then copy the file **data.pl** into your **bin** directory:

```
$ chmod 755 data.pl
$ cp data.pl bin
```

Then, execute the script:

```
$ data.pl
```

Everything after the `__END__` sequence within the file should print.

1. Using `vi bin/data.pl`, replace the `print if` line to read:

```
s/data/xxxx/;
print;
```

What happens?

2. Using `vi bin/data.pl`, replace the `print if` line to read:

```
s/the/THE/;
print;
```

What happens?

3. Using `vi bin/data.pl`, replace the `print if` line to read:

```
s/the/THE/g;
print;
```

What happens?

4. Using `vi bin/data.pl`, replace the `print if` line to read:

```
tr/data/xxxx/;  
print;
```

What happens?

5. Create a new file called `rot13.pl` with the following contents:

```
#!/usr/bin/perl  
  
while (<>)  
{  
    tr/A-Za-z/N-ZA-Mn-za-m/;  
    print;  
}
```

Change the executable status of the file, then copy the file `rot13.pl` into your `bin` directory:

```
$ chmod 755 rot13.pl  
$ cp rot13.pl bin
```

Execute this command:

```
$ rot13.pl rot13.pl
```

Execute the command again, this time piping the output to a file called `output`:

```
$ rot13.pl rot13.pl > output
```

Display the contents of the file:

```
$ cat output
```

Issue the following command:

```
$ rot13.pl output
```

If you received the same file that you entered (i.e., `rot13.pl`), then - congratulations - you are done!