

1. Write a Perl program `./merge_whales.pl` which reads a list of whales observations from its standard input and prints them to its standard output, merging adjacent counts of the same species. For example, given this input:

```
3 beluga
10 humpback
15 humpback
2 sperm whale
3 humpback
5 humpback
8 humpback
1 beluga
3 beluga
```

Your program should produce this output:

```
3 beluga
25 humpback
2 sperm whale
16 humpback
4 beluga
```

2. Write a Perl program, `phone_numbers.pl` which given the URL of a web page fetches it by running `wget` and prints any strings that might be phone numbers in the web page.
Assume the digits of phone numbers may be separated by zero or more spaces or hyphens('-') and can contain between 8 and 15 digits inclusive.

You should print the phone numbers one per line with spaces & hyphens removed.

For example

```
$ ./phone_numbers.pl http://www.unsw.edu.au
20151028
11187777
841430912571345
413200225
61293851000
57195873179
```

3. What does each of the following Perl code fragments print (no, don't just clip the lines and pass them to Perl, think about what they're doing):

```
$x = 'x';
for ($i = 1; $i <= 3; $i++) {
    $x = "($x)";
}
print "$x\n";
```

```
@hi = split //,"hello";
for ($i = 0; $i < 4; $i++) {
    print $hi[$i];
}
print "\n";
```

```
@vec = (10 .. 20);
print "@vec[1..3]\n";
```

```
foreach $n (1..10) {
    last if ($n > 5);
    print "$n ";
    next if ($n % 2 == 0);
    print "$n ";
}
print "\n";
```

4. Write a Perl program `source_count.pl` which prints the number of line of C source code in the current directory. In other words, this Perl program should behave similarly to `wc -l *.c`. (Note: you are not allowed to use `wc` or other Unix programs from within the Perl script). For example:

```
$ ./source_count.pl
  383 cyclorana.c
  280 cyclorana.h
   15 enum.c
  194 frequency.c
  624 model.c
  293 parse.c
  115 random.c
   51 smooth.c
  132 util.c
   16 util.h
  410 waveform.c
2513 total
```

5. Write a Perl program, `cut.pl` which takes three arguments, n , m and a file name. It should print characters n - m of each line of the file. For example:

```
$ ./cut.pl 1 8 file
```

should print the 8 characters of every line in `file`.

Implement a version of the program which invokes `/usr/bin/cut` and a version which performs the operations directly in Perl.

6. Implement a Perl script to solve the marks-to-grades problem that was solved as a shell script in a previous tutorial. Reminder: the script reads a sequence of (studentID,mark) pairs from its standard input and writes (studentID,grade) pairs to its standard output. The input pairs are written on a single line, separated by spaces, and the output should use a similar format. The script should also check whether the second value on each line looks like a valid grade, and output an appropriate message if it doesn't. The script can ignore any extra data occurring after the mark on each line. Consider the following input and corresponding output to the program:

Input

```
2212345 65
2198765 74
2199999 48
2234567 50 ok
2265432 99
2121212 hello
2222111 120
2524232 -1
```

Output

```
2212345 CR
2198765 CR
2199999 FL
2234567 PS
2265432 HD
2121212 ?? (hello)
2222111 ?? (120)
2524232 ?? (-1)
```

7. Write a program `addressbook.pl` that reads two files `people.txt` and `phones.txt` containing data in CSV (comma-separated values) format and uses this data to print an address book in the format below:

```
$ cat people.txt
andrew,Andrew Taylor,32 Lovely St,Glebe
arun,Arun Sharma,94 Leafy Close,Brisbane
gernot,Gernot Heiser,64 Trendy Tce,Newtown
jas,John Shepherd,16/256 Busy Rd,Alexandria
$ cat phones.txt
jas,home,9665 6432
arun,work,9385 5518
jas,work,9385 6494
arun,home,(07) 9314 6543
andrew,work,9385 4321
arun,mobile,0803 123 432
$ ./addressbook.pl
Andrew Taylor
32 Lovely St, Glebe
Phones: 9385 4321 (work)

Arun Sharma
94 Leafy Close, Brisbane
Phones: 0803 123 432 (mobile), (07) 9314 6543 (home), 9385 5518 (work)

Gernot Heiser
64 Trendy Tce, Newtown
Phones: ?

John Shepherd
16/256 Busy Rd, Alexandria
Phones: 9665 6432 (home), 9385 6494 (work)
```

Assume that there are only three types of phone (mobile, home and work) and we always display them in that order.

Hint: because the phone types are fixed, login name and phone type together can be the key used to look up a number. In this situation some suitable separator is used to create a composite, unambiguous key for the hash.