Week 08 Weekly Test Questions

Test Conditions

These questions must be completed under self-administered exam-like conditions. You must time the test yourself and ensure you comply with the conditions below.

- You may complete this test in CSE labs or elsewhere using your own machine.
- You may complete this test at any time before **Thursday 30 July 21:00**.
- Weekly tests are designed to act like a past paper - to give you an idea of how well you are progressing in the course, and what you need to work on. Many of the questions in weekly tests are from past final exams.
- Once the first hour has finished, you must submit all questions you've worked on.
- You should then take note of how far you got, which parts you didn't understand.
- You may choose then to keep working and submit test question anytime up to Thursday 30 July 21:00
- However the maximum mark for any question you submit after the first hour will be 50%

You may access this **language documentation** while attempting this test:

- Shell/Regex/Perl quick reference
- full Perl documentation

You may also access manual entries (the `man` command).

Any violation of the test conditions will results in a mark of zero for the entire weekly test component.

Set up for the test by creating a new directory called `test08`, changing to this directory, and fetching the provided code by running these commands:

```
$ mkdir test08
$ cd test08
$ 2041 fetch test08
```

Or, if you're not working on CSE, you can download the provided code as a zip file or a tar file.

Test Complete!

Your time for this test has finished. You must submit your work now. You should reflect on how you went in this hour, and discuss with your tutor if you have concerns. You may choose to keep working, but the maximum mark for any questions you submit later will be 50%.

WEEKLY TEST QUESTION:

Print Your Median Argument

Write a Perl program **median_number.pl** that given positive integers as command line arguments prints the median (middle) value.

Your program can assume is given an odd number of arguments.

Your program can assume all its arguments are positive integers.

For example:

```
$ ./median_number.pl 1 333 42
42
$ ./median_number.pl 3 4 2 1 7 6 5
4
$ ./median_number.pl 15 15 8 11 8
11
$ ./median_number.pl 42 42 244 244 42
42
```

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

```
$ 2041 autotest median_number
```

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

```
$ give cs2041 test08_median_number median_number.pl
```

WEEKLY TEST QUESTION:

List Identical Files in Shell

Write a Shell program, **ls_identical.sh** which takes the pathnames of 2 directories as argument.

It should print in alphabetical order the names of all files which occur in both directories and have exactly the same contents.

Files must have the same name in both directories and the same contents for their name to be printed.

Do not print the names of files with same contents but different names in both directories.

For example:

```
$ ls_identical.1.sh directory1 directory2
```

You can assume file names do not start with '.'.

Your answer must be Shell. You can not use other languages such as Perl, Python or C.

No error checking is necessary.

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

```
$ 2041 autotest shell_ls_identical
```

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

```
$ give cs2041 test08_shell_ls_identical ls_identical.sh
```

WEEKLY TEST QUESTION:

List Identical Files in Perl

Write a Perl program, **ls_identical.pl** which takes the pathnames of 2 directories as argument.

It should print in alphabetical order the names of all files which occur in both directories and have exactly the same contents.

Files must have the same name in both directories and the same contents for their name to be printed.

Do not print the names of files with same contents but different names in both directories.

For example:

```
$ mkdir directory1 directory2
$ echo hello >directory1/same.txt
$ echo hello >directory2/same.txt
$ echo hello >directory1/different.txt
$ echo world >directory2/different.txt
$ echo hello >directory1/one.txt
$ echo hello >directory2/two.txt
$ touch directory1/empty.txt directory2/empty.txt
$ ls directory1
different.txt
empty.txt
one.txt
same.txt
$ ls directory2
different.txt
empty.txt
same.txt
two.txt
$ ls_identical.pl directory1 directory2
empty.txt
same.txt
```

You can assume file names do not start with '.'.

Your answer must be Perl only. You can not use other languages such as Shell, Python or C.

You may not run external programs, e.g. via system or backquotes. for example, you can't run `diff`.

You may use any Perl module installed on CSE systems.

No error checking is necessary.

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

```
$ 2041 autotest perl_ls_identical
```

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

```
$ give cs2041 test08_perl_ls_identical ls_identical.pl
```

Submission

When you are finished each exercise 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 **Thursday 30 July 21:00** to complete this test.

Automarking will be run by the lecturer several days after the submission deadline for the test, using test cases that you haven't seen: different to the test cases `autotest` runs for you.

(Hint: do your own testing as well as running `autotest`)

Test Marks

After automarking is run by the lecturer you can view it here the resulting mark will also be available via via give's web interface or by running this command on a CSE machine:

```
$ 2041 classrun -sturec
```

The test exercises for each week are worth in total 1 marks.

The best 6 of your 8 test marks for weeks 3-10 will be summed to give you a mark out of 9.
