Week 04 Weekly Test Sample Answers

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 02 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 02 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 `test04`, changing to this directory, and fetching the provided code by running these commands:

```
$ mkdir test04
$ cd test04
$ 2041 fetch test04
```

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

WEEKLY TEST QUESTION:

Create A File of Integers In Shell

Write a Shell program, `create_integers_file.sh` which takes 3 arguments.

The first & second arguments will specify a range of integers.

The third argument will specify a filename.

Your program should create a file of this name containing the specified integers.

For example:

```
 ./create_integers_file.sh 40 42 fortytwo.txt
 cat fortytwo.txt
40
41
42
 ./create_integers_file.sh 1 5 a.txt
 cat a.txt
1
2
3
4
5
 ./create_integers_file.sh 1 1000 1000.txt
 wc 1000.txt
1000 1000 3893 1000.txt
```

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

You are not permitted to use the Linux program **seq**.

No error checking is necessary.

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

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

```
$ 2041 autotest shell_create_integers_file
```

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

```
$ give cs2041 test04_shell_create_integers_file create_integers_file.sh
```

Sample solution for `create_integers_file.sh`

```
#!/bin/bash

start=$1
finish=$2
filename="$3"

i=$start
while ((i <= finish))
do
    echo $i
    i=$((i + 1))
done >$filename
```

WEEKLY TEST QUESTION:

Create A File of Integers In Perl

Write a Perl program, `create_integers_file.pl` which takes 3 arguments.

The first & second arguments will specify a range of integers.

The third argument will specify a filename.

Your program should create a file of this name containing the specified integers.

For example:

```
$ ./create_integers_file.pl 40 42 fortytwo.txt
$ cat fortytwo.txt
40
41
42
$ ./create_integers_file.pl 1 5 a.txt
$ cat a.txt
1
2
3
4
5
$ ./create_integers_file.pl 1 1000  1000.txt
$ wc 1000.txt
1000 1000 3893 1000.txt
```

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.

No error checking is necessary.

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

```
$ 2041 autotest perl_create_integers_file
```

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

```
$ give cs2041 test04_perl_create_integers_file create_integers_file.pl
```

Sample solution for `create_integers_file.pl`

```perl
#!/usr/bin/perl -w

# create a file containing numbers min..max 1 per line
# written by andrewt@unsw.edu.au as COMP[29]041 sample solution

die "Usage: $0 <min> <max> <file>\n" if @ARGV != 3;

$min = $ARGV[0];
$max = $ARGV[1];
$file = $ARGV[2];

open F, '>', $file or die "$0: ca not open file: $!\n";

foreach $i ($min..$max) {
    print F "$i\n";
}

close F;
```

Alternative solution for `create_integers_file.pl`

```perl
#!/usr/bin/perl -w

# create a file containing numbers min..max 1 per line
# written by andrewt@unsw.edu.au as COMP[29]041 sample solution
# more concise but less-readable solution

@ARGV == 3 and open F, '>', $ARGV[2] or die;
print F join("\n", ($ARGV[0]..$ARGV[1])), "\n";
```

WEEKLY TEST QUESTION:

Print the N-th Line of a File

Write a Perl program, `nth_line.pl` to print the *n*-th line of a file.

It will be given two arguments *n* and the file name.

Your program should print nothing if the file does not have an *n*-th line

You can assume *n* is a positive (non-zero) integer.

You should not assume anything about the lines in the file.

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.

No error checking is necessary.

```
./create_integers_file.sh 42 99 numbers.txt
head numbers.txt
42
43
44
45
46
47
48
49
50
51
tail numbers.txt
90
91
92
93
94
95
96
97
98
99
./nth_line.pl 1 numbers.txt
42
./nth_line.pl 20 numbers.txt
61
./nth_line.pl 1000 numbers.txt
echo this file has one line >file.txt
cat file.txt
this file has one line
./nth_line.pl 1 file.txt
this file has one line
./nth_line.pl 42 file.txt
```

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

```
$ 2041 autotest nth_line
```

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

```
$ give cs2041 test04_nth_line nth_line.pl
```

Sample solution for `nth_line.pl`

```
#!/usr/bin/perl -w

# print nth-line of a file
# written by andrewt@unsw.edu.au as COMP[29]041 sample solution

die "Usage $0: <n> <file>\n" if @ARGV != 2;

open F, "<", $ARGV[1] or die "$0: can not open $ARGV[1]: $!\n";
$line_number = 1;
while ($line = <F>) {
    if ($line_number == $ARGV[0]) {
        print $line;
        last;
    }
    $line_number++;
}
```

Alternative solution for `nth_line.pl`

```
#!/usr/bin/perl -w

# print nth-line of file
# written by andrewt@unsw.edu.au as COMP[29]041 sample solution
# more concise but less readable solution

$target_line_number = shift @ARGV or die "Usage $0: <n> <file>\n";
$. == $target_line_number and print while <>;
```

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 02 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.
