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 **Wednesday 05 September 23:59:59**
- The maximum time allowed for this test is 1 hour + 5 minutes reading time.
- You may first use 5 minutes to read the questions (no typing)
- You then must complete the test within 1 hour and submit your answers with give.
- You must complete the questions alone - you can not get help in any way from any person.
- You can not access your previous answers to lab or tut questions.
- You can not access web pages or use the internet in any way.
- You can not access books, notes or other written or online materials.
- You can not access your own files, programs, code ...
- You can not access COMP2041 course materials except for language documentation linked below.

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

- Shell/Regex/Perl quick reference
- Javascript quick reference
- full Perl documentation
- Python quick reference
- C quick reference
- full Python 3.6 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.

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:

```
$ 2041 autotest shell_create_integers_file
```

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

```
$ give cs2041 test06_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
```

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 test06_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";
```

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 test06_nth_line nth_line.pl
```

Sample solution for `nth_line.pl`

```perl
#!/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;
        exit 1;
    }
    $line_number++;
}
exit 0;
```

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 **Wednesday 05 September 23:59:59** 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`)

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