Example_Module.pm

```
package Example_Module;
# written by andrewt@cse.unsw.edu.au for COMP2041
#
# Definition of a simple Perl module.
#
# List::Util provides the functions below and more

use base 'Exporter';
our @EXPORT = qw/sum min max minstr maxstr/;
use List::Util qw/reduce/;


sub sum {
    return reduce {$a + $b} @_;
}

sub min {
    return reduce {$a < $b ? $a : $b} @_;
}

sub max {
    return reduce {$a > $b ? $a : $b} @_;
}

sub minstr {
    return reduce {$a lt $b ? $a : $b} @_;
}

sub maxstr {
    return reduce {$a gt $b ? $a : $b} @_;
}

# necessary
1;
```

module_example.pl

Use of a simple Perl module.

```
use Example_Module qw/max/;

# As max is specified in our import list it can be used without the package name
print max(42,3,5), "\n";

# We don't import min explicitly so it needs the package name
print Example_Module::min(42,3,5), "\n";
```

test_legit.pl

Run legit.pl on a series of commands and run the reference implementation (2041 legit) in parallel on the same commands
stop with an error if legit.pl's behaviour differences from the reference implementation
use some useful functions from Perl core modules

```
use File::Temp;
use Cwd;

# create two temporary directories
# this is done conveniently and securely by File::Temp
# File::Temp also automatically removes the directory when the program finishes

$test_directory = File::Temp->newdir or die;
$reference_directory = File::Temp->newdir or die;

# read commands and execute them
sub main {
    my $command;
    while ($command = <>) {
        chomp $command;
        execute($command) if $command;
    }
    print "*** Test passed ***\n";
    exit 0;
}

# execute a command using legit.pl
# and using the refenece implementation
# then check results are identical

sub execute {
    my ($command) = @_;
    print "\$ $command\n";

    # because of the cd - replace legit.pl with its full pathname
    my $test_command = $command;
    my $current_directory = getcwd;
    $test_command =~ s/(.\/)?legit\.pl/$current_directory\/legit.pl/g;

    my $test_output = `cd $test_directory; $test_command 2>&1`;
    print $test_output;

    # replace legit.pl with 2041 legit
    my $reference_command = $command;
    $reference_command =~ s/(.\/)?legit\.pl/2041 legit/g;

    my $reference_output = `cd $reference_directory; $reference_command 2>&1`;
```