Exercises Chapter 1

ex1-3.

```
#! /usr/bin/perl
## Copyright (C) 2013 by Ma
@lines = `perldoc -u -f atan2`;
foreach (@lines) {
    s/\w<([^>]+)>/\U$1/g;
    print;
}
```

Exercises Chapter 2

ex2-1.

```
#! /usr/bin/perl
## Copyright (C) 2013 by Ma

$pi = 3.141592654;
print "The circumference of a circle with a radius of 12.5 is ". 2 * $pi * 12.5 ."\n";
```

ex2-2.

```
#! /usr/bin/perl
## Copyright (C) 2013 by Ma

$pi = 3.141592654;
print "What is the radius? ";
chomp($radius = <STDIN>);
print "The circumference of a circle with a radius of 12.5 is ". 2 * $pi * $radius ."\n";
```

ex2-3.

```
#! /usr/bin/perl
## Copyright (C) 2013 by Ma

$pi = 3.141592654;
print "What is the radius? ";
chomp($radius = <STDIN>);
if ($radius < 0) {
    $radius = 0;
}
print "The circumference of a circle with a radius of 12.5 is ". 2 * $pi * $radius ."\n";
```

ex2-4.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
```

```
    print "Please Enter a number: ";
    chomp($a = <STDIN>);
    print "Please Enter another: ";
    chomp($b = <STDIN>);
    $result = $a * $b;
    print "$a multiplys $b equals $ result.\n"
```

ex2-5.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    print "Please Enter a string: ";
    chomp($str = <STDIN>);
    print "Please a number: ";
    chomp($times = <STDIN>);
    $result = $str x $times;
    print "The result is $result.\n"
```

Exercises Chapter 3

ex3-1.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    print "Please some lines and end with Ctrl+D\n";
    @lines = <STDIN>;
    @lines = reverse @lines;
    print "The reverse lines is:\n";
    print @lines;
```

ex3-2.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    print "Please enter some numbers, one per line, and end with Ctrl+D.\n";
    @names = qw {fred betty barney dino wilma pebbles bamm-bamm};
    chomp(@numbers = <STDIN>);
    print "The result is:\n";
    foreach (@numbers) {
       print "$names[$_ -1]\n";
    }
```

ex3-3.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    print "Please enter some strings, one per line, and end with Ctrl+D.\n";
    chomp(@lines = <STDIN>);
```

```
    @sort_lines = sort @lines;
    print "Output result on one line:\n";
    print "@sort_lines\n";

    print "Output result on separate lines:\n";
    foreach (@sort_lines) {
       print "$_\n";
    }
```

Exercises Chapter 4

ex4-1.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    sub total {
       my $sum = 0;
       foreach(@_) {
          $sum += $_;
       }
       $sum;
    }

    my @fred = qw{ 1 3 5 7 9 };
    my $fred_total = total(@fred);
    print "The total of \@fred is $fred_total.\n";
```

ex4-2.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    sub total {
       my $sum = 0;
       foreach(@_) {
          $sum += $_;
       }
       $sum;
    }

    print "The sum of the numbers from 1 to 1000 is: ", total(1..1000), ".\n";
```

ex4-3.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    sub average {
       my $sum = 0;
```

```
    foreach (@_) {
        $sum += $_;
    }
    $sum / @_;
}

sub above_average {
    my $average = average(@_);
    my @result;
    foreach (@_) {
        if ($_ > $average) {
            push @result, $_;
        }
    }
    @result;
}

my @fred = above_average(1..10);
print "\@fred if @fred\n";
print "(Should be 6 7 8 9 10)\n";
my @barney = above_average(100, 1..10);
print "\@barney is @barney\n";
print "(Should be just 100)\n";
```

ex4-4.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
use 5.010;
sub greet {
    state $last_person;
    my $person = shift @_;
    if ( defined $last_person) {
        print "Hi $person! $last_person is also here!\n";
    } else {
        print "Hi $person! You are the first one here!\n";
    }
    $last_person = $person;
}

greet('Fred');
greet('Barney');
```

ex4-5.

```
#!/usr/bin/perl
```

```
## Copyright (C) 2013 by Ma
use 5.010;

sub greet {
    state @person_list;
    my $person = shift @_;

    if (@person_list != 0) {
        print "Hi $person! I've seen: @person_list \n";
    } else {
        print "Hi $person! You are the first one here!\n";
    }
    push @person_list, $person
}

greet("Fred");
greet("Barney");
greet("Wilma");
greet("Betty");
```

Exercises Chapter 5

ex5-1.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
print reverse <>;
```

ex5-2.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
print "Please enter some strings, end with Ctrl+D:\n";
chomp(my @strings = <STDIN>);
print "1234567890" x 3, "\n";
printf "%20s\n" x @strings, @strings;
```

ex5-3.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
print "What column width do you want: ";
chomp(my $width = <STDIN>);

print "Please enter some strings, end with Ctrl+D:\n";
chomp(my @strings = <STDIN>);
```

```
    print "1234567890" x ($width/9 + 2), "\n";
    printf "%${width}s\n" x @strings, @strings;
```

Exercises Chapter 6

ex6-1.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    my %names = (
       fred => 'flintstone',
       barney => 'rubble',
       wilma => 'fintstione'
    );
    print "Please enter a first name: ";
    chomp(my $name = <STDIN>);
    print "${name}'s last name is $names{$name}.\n";
```

ex6-2.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    print "Please enter some words, end with Ctrl+D.\n";
    chomp(my @words = <STDIN>);

    my %count;
    foreach (@words) {
       $count{$_} += 1;
    }

    foreach my $word (sort keys %count) {
       print "$word was seen $count{$word} times.\n";
    }
```

ex6-3.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    my $max = 0;
    while (($key, $value) = each %ENV) {
       my $length = length($key);
       if ($length > $max) {
          $max = $length;
       }
    }

    while (($key, $value) = each %ENV) {
```

```
        printf "%-${max}s    %s\n", $key, $value;
    }
```

Exercises Chapter 7

ex7-1.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    while(<>){
        if (/fred/) {
            print;
        }
    }
```

ex7-2.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    while(<>){
        if (/[fF]red/) {
            print;
        }
    }
```

ex7-3.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    while(<>){
        if (/\./) {
            print;
        }
    }
```

ex7-4.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
    while(<>){
        if (/^[A-Z][A-Z]*[a-z]+[A-Za-z]*/) {
            print;
        }
    }
```

ex7-5.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma
```

```
while(<>){
    if (/(\S)\g{-1}/a) {
        print;
    }
}
```

ex7-6.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
while(<>){
    if (/wilma.*fred|fred.*wilma/) {
        print;
    }
}
```

Exercises Chapter 8

ex8-1.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
while (<>) {                        # take one input line at a time
    chomp;
    if (/match/) {
        print "Matched: |$`<$&>$'|\n";    # the special match vars
    } else {
        print "No match: |$_|\n";
    }
}
```

ex8-2.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
while (<>) {                        # take one input line at a time
    chomp;
    if (/\b\w*a\b/) {
        print "Matched: |$`<$&>$'|\n";    # the special match vars
    } else {
        print "No match: |$_|\n";
    }
}
```

ex8-3.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
```

```
   while (<>) {                          # take one input line at a time
      chomp;
      if (/(\b\w*a\b)/) {
         print "\$1 contains '$1'\n";    # the special match vars
      } else {
         print "No match: |$_|\n";
      }
   }
```

ex8-4.

```
   #!/usr/bin/perl
   ## Copyright (C) 2013 by Ma
   use 5.010;

   while (<>) {                          # take one input line at a time
      chomp;
      if (/(?<word>\b\w*a\b)/) {
         print "'word' contains '${word}'\n";    # the special match vars
      } else {
         print "No match: |$_|\n";
      }
   }
```

ex8-5.

```
   #!/usr/bin/perl
   ## Copyright (C) 2013 by Ma
   use 5.010;

   while (<>) {                          # take one input line at a time
      chomp;
      if (/(?<word>\b\w*a\b)(?<rest>[\s\S]{0,5})/) {
         print "'word' contains '${word}'. The 'rest' is '${rest}'.\n";    # the special match vars
      } else {
         print "No match: |$_|\n";
      }
   }
```

ex8-6.

```
   #!/usr/bin/perl
   ## Copyright (C) 2013 by Ma
   while (<>) {                          # take one input line at a time
      chomp;
      if (/\s\z/) {
         print "$_***\n";    # the special match vars
```

```
    }
}
```

Exercises Chapter 9

ex9-1.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
$what = "fred";
while (<>) {                            # take one input line at a time
    chomp;
    if (/($what){3}/) {
        print "Matched: |$`<$&>$'|\n";    # the special match vars
    } else {
        print "No match: |$_|\n";
    }
}
```

ex9-2.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
my $filename_in = $ARGV[0];

my $filename_out =$filename_in;

$filename_out =~ s/(\.\w+)?$/.out/;

if (! open $f_in, '<', $filename_in) {
    die "'$filename_in': $!";
}

if (! open $f_out, '>', $filename_out) {
    die "'$filename_out': $!";
}

while (<$f_in>) {
    s/Fred/Larry/gi;
    print $f_out $_;
}
```

ex9-3.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma
my $filename_in = $ARGV[0];
```

```
    my $filename_out =$filename_in;

    $filename_out =~ s/(\.\w+)?$/.out/;

    if (! open $f_in, '<', $filename_in) {
        die "'$filename_in': $!";
    }

    if (! open $f_out, '>', $filename_out) {
        die "'$filename_out': $!";
    }

    while (<$f_in>) {
        chomp;
        s/Fred/\n/gi;
        s/Wilma/Fred/gi;
        s/\n/Wilma/g;
        print $f_out "$_\n";
    }
```

ex9-4.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma

    $^I = ".bak";
    while (<>) {
        if (/\A#!/) {
            $_ .= "## Copyright (C) 2013 by Ma\n";
        }
        print;
    }
```

ex9-5.

```
    #!/usr/bin/perl
    ## Copyright (C) 2013 by Ma

    my %todo_list;
    foreach (@ARGV) {
        $todo_list{$_} = 1;
    }

    while (<>) {
        if (/\A## Copyright/) {
```

```
            delete $todo_list{$ARGV};
        }
    }

    @ARGV = sort keys %todo_list;

    if (@ARGV) {        #do next, if files exists.
        $^I = ".bak";
        while (<>) {
            if (/\A#!/) {
                $_ .= "## Copyright (C) 2013 by Ma\n";
            }
            print;
        }
    }
```

Exercises Chapter 10

ex10-1.

```
    #!/usr/bin/perl

    my $number = int(1+rand 100);

    while (1) {
        print "Please Enter a number from 1 to 100: ";
        chomp ($num = <STDIN>);
        if ($num =~ /quit|exit|\A\s*\z/i) {
            print "Now to exit!\n";
            last;
        } elsif ($num < $number) {
            print "Too small. Try again.\n";
        } elsif ($num > $number) {
            print "Too large. Try again.\n";
        } else {
            print "You guess!\n";
            last;
        }
    }
```

ex10-2.

```
    #!/usr/bin/perl

    use 5.010;
```

```
my $debug = $ARGV[0] // 0;
my $number = int(1+rand 100);

print "We are in debug mode, the number is $number.\n" if $debug;

while (1) {
   print "Please Enter a number from 1 to 100: ";
   chomp ($num = <STDIN>);
   if ($num =~ /quit|exit|\A\s*\z/i) {
      print "Now to exit!\n";
      last;
   } elsif ($num < $number) {
      print "Too small. Try again.\n";
   } elsif ($num > $number) {
      print "Too large. Try again.\n";
   } else {
      print "You guess!\n";
      last;
   }
}
```

ex10-3.

```
#!/usr/bin/perl
## Copyright (C) 2013 by Ma

use 5.010;
$ENV{ZERO} = 0;
$ENV{EMPTY} = '';
$ENV{UNDEFINED} = undef;

my $max = 0;
while (($key, $value) = each %ENV) {
   my $length = length($key);
   if ($length > $max) {
      $max = $length;
   }
}

while (($key, $value) = each %ENV) {
   printf "%-${max}s    %s\n", $key, $value // "(undefined)";
}
```

Exercises Chapter 11

ex11-1.

```
#!/usr/bin/perl
use Module::CoreList;

my %modules = %{$Module::CoreList::version{5.014}};

foreach (keys %modules) {
    print "$_\n";
}
```

ex11-2.

```
#!/usr/bin/perl

use DateTime;
use Time::Piece;

my $t = localtime;

my $now = DateTime->new (
    year => $t->year,
    month => $t->mon,
    day => $t->mday,
  );

my $input_date = DateTime->new (
    year => $ARGV[0],
    month => $ARGV[1],
    day => $ARGV[2],
);

my $duration = $now - $input_date;

my @units = $duration->in_units( qw(years months days));

printf "%d years, %d months, and %d dyas.\n",@units;
```

Exercises Chapter 12

ex12-1.

```
#!/usr/bin/perl
foreach my $file (@ARGV) {
    if (-e $file) {
        print "$file is";
        print " readable" if -r $file;
```

```
            print " writable" if -w $file;
            print " executable" if -x $file;
            print ".\n";
        } else {
            print "$file does not exist.\n";
        }

    }
```

ex12-2.

```
    #!/usr/bin/perl
    die "No file input!\n" unless @ARGV;
    my $name = shift @ARGV;
    my $oldest_age = -C $name;

    foreach (@ARGV) {
        my $age = -C;
        ($name, $oldest_age) = ($_, $age)
              if $age > $oldest_age;
    }

    printf "The oldest file is %s, it is %f days old.\n",
          $name, $oldest_age;
```

ex12-3.

```
    #!/usr/bin/perl
    use 5.010;

    die "No file input!\n" unless @ARGV;

    foreach (@ARGV) {
        print "$_ is owned by your and readable, writable.\n"
            if -o -r -w $_;
    }
```

Exercises Chapter 13

ex13-1.

```
    #!/usr/bin/perl
    print "Enter a directory: ";
    chomp (my $dir = <STDIN>);
    if ($dir =~ /\A\s*\Z/) {
        chdir or die "Can't chdir to you home: $!.\n";
    } else {
```

```
    chdir $dir or die "Can't chdir to $dir: $!.";
}

foreach (sort <*>) {
    print "$_\n";
}
```

ex13-2.

```
#!/usr/bin/perl
print "Enter a directory: ";
chomp (my $dir = <STDIN>);
if ($dir =~ /\A\s*\Z/) {
    chdir or die "Can't chdir to you home: $!.\n";
} else {
    chdir $dir or die "Can't chdir to $dir: $!.";
}

foreach (sort <.* *>) {
    print "$_\n";
}
```

ex13-3.

```
#!/usr/bin/perl
print "Enter a directory: ";
chomp (my $dir = <STDIN>);
if ($dir =~ /\A\s*\Z/) {
    chdir or die "Can't chdir to you home: $!.\n";
} else {
    chdir $dir or die "Can't chdir to $dir: $!.";
}

opendir DIR, "." or die "Can't opendir $!";
foreach (sort readdir DIR) {
    print "$_\n";
}
```

ex13-4.

```
#!/usr/bin/perl
unlink @ARGV;
```

ex13-5.

```
#!/usr/bin/perl
use File::Basename;
use File::Spec;
```

```
    my($src, $dest) = @ARGV;

    if (-d @dest) {
        my $basename = basename $src;
        $dest = FIle::Spec->catfile($dest, $basename);
    }

    rename $src, $dest or die "Can't rename '$src' to '$dest'.\n";
```

ex13-6.

```
    #!/usr/bin/perl
    use File::Basename;
    use File::Spec;

    my($src, $dest) = @ARGV;

    if (-d @dest) {
        my $basename = basename $src;
        $dest = FIle::Spec->catfile($dest, $basename);
    }

    link $src, $dest or die "Can't rename '$src' to '$dest'.\n";
```

ex13-7.

```
    #!/usr/bin/perl
    use File::Basename;
    use File::Spec;

    my $symlink = $ARGV[0] eq '-s';
    shift @ARGV if $symlink;

    my($src, $dest) = @ARGV;

    if (-d @dest) {
        my $basename = basename $src;
        $dest = FIle::Spec->catfile($dest, $basename);
    }

    if ($symlink) {
        symlink $src, $dest or die "Can't rename '$src' to '$dest'.\n";
    } else {
        link $src, $dest or die "Can't rename '$src' to '$dest'.\n";
    }
```

ex13-8.

```
#!/usr/bin/perl
foreach (<.* *>) {
    my $dest = readlink $_;
    print "$_ -> $dest\n" if defined $dest;
}
```

Exercises Chapter 14

ex14-1.

```
#!/usr/bin/perl

print "Enter some numbers, one per line.\n";

my @numbers;

while (<>) {
    push @numbers, $_;
}
@numbers = sort {$a <=> $b} @numbers;

foreach (@numbers) {
    printf "%10g\n", $_;
}
```

ex14-2.

```
#!/usr/bin/perl

my %last_name = qw{
    fred flintstone Wilma Flintstone Barney Rubble
    betty rubble Bamm-Bamm Rubble PEBBLES FLINTSTONE
};

my @result = sort {
    "\L$last_name{$a}" cmp "\L$last_name{$b}"
        or
    "\L$a" cmp "\L$b"
} keys %last_name;

foreach (@result) {
    print "$_ $last_name{$_}\n";
}
```

ex14-3.

```
#!/usr/bin/perl
print "Enter a String: ";
chomp (my $string = <STDIN>);
print "Enter a substring: ";
chomp (my $substring = <STDIN>);


my $index = -1;

while (1) {
   $index = index($string, $substring, $index + 1);
   if ($index == -1) {
      last;
   } else {
      print "$index\n";
   }
}
```

Exercises Chapter 15

ex15-1.

```
#!/usr/bin/perl
use 5.014;

my $number = int(1+rand 100);

while (1) {
   print "Please Enter a number from 1 to 100: ";
   chomp (my $num = <STDIN>);
   given ($num) {
      when (! /\A\d+\Z/) {
         say "Not a number!"}
      when ($_ > $number) {
         say "Too large. Try again."}
      when ($_ < $number) {
         say "Too small. Try again."}
      when ($_ == $number) {
         say "You guess!"; last
      }
   }
}
```

ex15-2.

```
#!/usr/bin/perl

use 5.014;

print "Enter a number: ";
chomp (my $num = <STDIN>);

given ($num) {
   when ($_ % 3 == 0) { say "Fizz"; continue}
   when ($_ % 5 == 0) { say "Bin"; continue}
   when ($_ % 7 == 0) { say "Sausage"; }
}
```

ex15-3.

```
#!/usr/bin/perl
use 5.014;

for (@ARGV) {
   say "\nProcessing $_";

   when (! -e) { say " File does not exist."}
   when (-r _) { say " Readable."; continue}
   when (-w _) { say " Writable."; continue}
   when (-x _) { say " Executable."}
}
```

ex15-4.

```
#!/usr/bin/perl

use 5.014;
sub divisors {
   my $number = shift;

   my @divisors = ();
   foreach my $divisor (2 .. ($number/2)) {
      push @divisors, $divisor unless $number % $divisor;
   }

   return @divisors;
}
print "Enter a num: ";
chomp (my $num = <STDIN>);
given ($num) {
```

```
    when (! /\A\d+\Z/) { say "Not a number!"}

    my @divisors = divisors($_);
    my @empty;
    when (@divisors ~~ @empty) { say "$_ is a prime"}
    default { say "$_ is divisible by @divisors"}
  }
```

ex15-5.

```
  #!/usr/bin/perl

  use 5.014;
  sub divisors {
    my $number = shift;

    my @divisors = ();
    foreach my $divisor (2 .. ($number/2)) {
      push @divisors, $divisor unless $number % $divisor;
    }

    return @divisors;
  }

  print "Enter a num: ";
  chomp (my $num = <STDIN>);
  given ($num) {
    when (! /\A\d+\Z/) { say "Not a number!"}

    my @divisors = divisors($_);

    when (2 ~~ @divisors || $_ == 2) {
      say "$_ is even";
      continue;
    }

    when ( !(2 ~~ @divisors) && $_ != 2) {
      say "$_ is odd";
      continue;
    }

    when (7 ~~ @divisors) {
      say "$_ is divisible by my favorite number 7";
      continue;
    }
```

```
        my @empty;
        when (@divisors ~~ @empty) { say "$_ is a prime"}

        default { say "$_ is divisible by @divisors"}
    }
```

Exercises Chapter 16

ex16-1.

```
    #!/usr/bin/perl

    chdir '/';
    system "ls -l";
```

ex16-2.

```
    #!/usr/bin/perl

    open STDOUT, '>', 'ls.out';
    open STDERR, '>', 'ls.err';

    chdir '/';
    system "ls -l";
```

ex16-3.

```
    #!/usr/bin/perl
    use 5.014;

    given (`date`) {
        when (/Mon|Tue|Wed|Thu|Fri/) {print "get to work!\n"}
        when (/Sun|Sat/) {print "go paly!\n"}
    }
```

ex16-4.

```
    #!/usr/bin/perl
    use 5.010;

    sub my_int_handler {
        print "Caught INT. Now exit.\n";
        exit;
    }
    sub my_usr1_handler {
        state $n;
        $n++;
```

```
    print "Caught USR1: $n\n";
  }
  print "I am $$.\n";

  $SIG{'INT'} = 'my_int_handler';
  $SIG{'USR1'} = 'my_usr1_handler';

  while (1) {};
```

Exercises Chapter 17

ex17-1.

```
  #!/usr/bin/perl

  my $filename = './sample_text.txt';
  open my $fn , '<', $filename or die "Can't open file:$filename!\n";

  my @strings;

  while (<$fn>) {
    chomp;
    push @strings, $_;
  }

  while (1) {
    print "Enter a pattern: ";
    chomp (my $pattern = <STDIN>);
    last if $pattern =~ /\A\s*\Z/;

    my @result = eval {
      grep /$pattern/, @strings;
    };

    if ($@) {
      print "Error: $@\n";
    } else {
      my $count = @result;
      print "There are $count matching strings:\n",
        join "\n", @result;
    }
    print "\n";
  }
```

ex17-2.

```
#!/usr/bin/perl

foreach (<*>) {
    my ($time_acess, $time_modify) = (stat)[8,9];
    printf "%-10s %10d %10d\n", $_, $time_acess, $time_modify;
}
```

ex17-3.

```
#!/usr/bin/perl

foreach (<*>) {
    my ($time_acess, $time_modify) = map {
        my($year, $month, $day) = (localtime($_))[5,4,3];
        $year += 1990;
        $month += 1;
        sprintf "%4d-%02d-%02d", $year, $month, $day;
    }(stat)[8,9];

    printf "%-10s %10s %10s\n", $_, $time_acess, $time_modify;
}
```