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

## Global variables
my $infile = $ARGV[0];  ## assign infile name
my %Fasta;              ## declaring hash we'll read fasta into


open(IN,"<$infile") or die "\n\nError cannot open the infile $infile\n\n";      ## open in infile or die
$/='>';                 ## change break character to ">"
my @FASTA = <IN>;       ## read the file into an array
close(IN);              ## close the file handle
shift(@FASTA);          ## Remove the first element in the array, because in this case it's empty

foreach my $sequence (@FASTA) {             ## iterating over each element of the array (each sequence of
                                             ##   the fasta file)
    my @seq_lines = split(/\n/, $sequence); ## creating a new sub-array which contains each of the lines
    my $header = $seq_lines[0];             ## header is always the first line, so the first element of this
                                             ##   sub-array will be the header
    my $seq_string = "";                    ## declaring and emptying the local variable seq_string
    foreach my $i (1..$#seq_lines) {        ## Iterating over the sub-array FROM THE SECOND element TO THE LAST
                                             ##   element of the array
        $seq_string = $seq_string . $seq_lines[$i]; ## tacking each line of the sequence lines together
    }                                       ## end loop
    $Fasta{$header} = $seq_string;          ## Assigning the header variable as the key to the value of its
                                             ##   respective sequence. Referencing the hash in the context of a
                                             ##   string, therefor use $ instead of %
}

## Print out the hash
foreach my $sequence (keys %Fasta) {        ## iterating over each element of the hash
    #my $key = $sequence;                   ## assigning local variable $key with the key of the hash
    #my $value = $Fasta{$sequence};         ## assigning local variable $value with value associated to the key
    print ">$sequence\n$Fasta{$sequence}\n";   ## print the key and value (in FASTA format) to standard out.
}
```

A Perl script that will read a FASTA file into a hash.
Note: Anything following a '#' is a comment.

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

my $fasta = $ARGV[0];           ## Input FASTA file
my $line_count = 0;             ## Line counter
my $header;                           ## Global variable that will at one point (briefly) hold each of the fasta headers.
my $sequence;                         ## Global variable that will at one point (briefly) hold each of the sequences.

open(IN,"<$fasta") || die "\n\n Cannot open the input FASTA file: $fasta\n\n";        ## open the FASTA file
while(<IN>) {                                                ## loop through the FASTA file, one line at a time.
    chomp;                                                   ## get rid of the \n at the end of each line
    if ($_ =~ m/^>/) {                                       ## if the line contains a greater than sign at the
                                                              ## beginning of the line
        if ($line_count == 0) {                              ## if line_count is equal to 0
            $header = $_;                                    ## the $header variable now equals the very first line
        }                                                    ## end loop
        else {                                               ## if line_count IS NOT EQUAL TO 0
            print "$header\n$sequence\n";                    ## print the header and the sequence variables
            $header = $_;                                    ## header now equals the line you are on
            $sequence = "";                                  ## sequence has been "cleared out" equals nothing
        }                                                    ## end loop
    }                                                        ## end loop
    else {                                                   ## if the line DOES NOT HAVE A greater than sign
                                                              ## at the beginning of the line
        $sequence = $sequence . $_;                          ## add this line to the variable sequence
    }                                                        ## end loop
    $line_count += 1;                                        ## add 1 to the variable line count. when the script
                                                              ## is done, line_count will equal the number of lines
                                                              ## infile

}                                                            ## end loop
close(IN);                                                   ## Close the file.
```

A Perl script that will flaten a FASTA file and print the output to the screen.
Note: Anything following a '#' is a comment.