Musterlösung zu Blatt 10:

2. Suchen Sie alle Überschriften im Text: (In html Dateien werden Überschriften mit den HTML-Tags <h1> ... </h1>, <h2> ... </h2> bis <h4>.. ausgezeichnet.)

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: findet Ueberschriften
use strict;
use utf8;
use locale;
{
  my ($zeile,$regex);
  open(TEXT,"<:utf8","polt.html") or die "file not found!";

  while($zeile = <TEXT>) {
    for(my $i=1;$i<=4;$i++) {
      $regex = "<h".$i.".*</h".$i.">";
      if($zeile =~ /$regex/) {
        print $zeile;
      }
    }
  }
  close TEXT;
}
```

3. Extrahieren Sie aus allen Überschriften den Wert des title Attributs

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: extrahiert title aus Ueberschriften

use strict;
use utf8;
use locale;
{
  my ($zeile,$regex);

  open(TEXT,"<:utf8","polt.html") or die "file not found!";

  while($zeile = <TEXT>) {
    for(my $i=1;$i<=4;$i++) {
      $regex = "<h".$i."(.*)</h".$i.">";
      if($zeile =~ /$regex/) {
        if($zeile =~ /title=\"(.*?)\"/) {
          print $1."\n";
        }
      }
    }
  }
  close TEXT;
}
```

4.Extrahieren Sie alle www Links und schreiben Sie die WWW-Adressen der Links in die Datei links.txt

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: extrahiert Links in links.txt

use strict;
use utf8;
use locale;
{
  my ($zeile);

  open(TEXT,"<:utf8","polt.html") or die "file not found!";
  open(LINKS,">:utf8","links.txt");

  while($zeile = <TEXT>) {
    if($zeile =~ /href=\"(.*?)\"/) {
      my $tmpzeile = $1;
      if($tmpzeile =~ /http/ || /www/ ) {
        print LINKS $tmpzeile."\n";
      }
    }
  }
  close TEXT;
  close LINKS;
}
```

5. Extrahieren Sie alle Zeilen, in denen "[Bearbeiten]" steht.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: extrahiert alle Zeilen in denen [Bearbeiten] steht

use strict;
use utf8;
use locale;
{
  my ($zeile);

  open(TEXT,"<:utf8","polt.html") or die "file not found!";

  while($zeile = <TEXT>) {
  #[Bearbeiten] wird in html als >Bearbeiten< dargestellt
    if($zeile =~ />Bearbeiten</) {
      print $zeile."\n";
    }
  }
  close TEXT;
}
```

6. Suchen Sie im Text alle Paare von identischen Wörtern. Groß/Kleinschreibung spielt keine Rolle.

Hinweis: Paare von identischen Wörtern bedeutet für mich etwas wie „die die“, also zweimal das selbe Wort hintereinander. Dies kommt im Text bisher nicht vor und muss zum Testen eingefügt werden.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: extrahiert alle Paare von identischen Woertern
use strict;
use utf8;
use locale;
{
  my ($zeile);
  open(TEXT,"<:utf8","polt.html") or die "file not found!";

  while($zeile = <TEXT>) {
    if ($zeile =~/\b(\p{L}+)\b \b(\1)\b/i) {
      print"$1 $2\n";
    }
  }
  close TEXT;
}
```

8./9. Erzeugen Sie eine Frequenzliste aller großgeschriebenen Wörter aus der Datei polt.txt, die länger als 5 Buchstaben sind: ACHTUNG: In den HASH dürfen nur Wörter eingetragen werden, die das Kriterium "großgeschrieben und länger als 5 Buchstaben" erfüllen. Wie kann mit Systemroutinen Aufrufen ermittelen, wieviele KEYS im HASH sind?

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: erstellt Frequenzliste aller grossen Woerter die laenger als fuenf
Buchstaben sind und gibt Anzahl der Keys aus

use strict;
use utf8;
use locale;
{
  my ($zeile,@woerter,%lexikon,@woerter_sortiert,$wort);
  open(TEXT, "<:utf8","polt.txt") or die "file not found!";

  while($zeile = <TEXT>) {
    @woerter = split(/[\p{Z}\p{P}\s]+/,$zeile);
    foreach $wort(@woerter){
      if($wort =~ /^\p{Lu}\p{L}{5,}$/) {
        $lexikon{$wort}++;
      }
    }
  }
```

```
  @woerter_sortiert = sort {$lexikon{$a} <=> $lexikon{$b}} keys %lexikon;

  foreach $wort(@woerter_sortiert){
    print "Das Wort \"$wort\" kommt $lexikon{$wort}mal im Text vor.\n";
  }

  my $anzahl = scalar(keys %lexikon);
  print "Im Text sind $anzahl Keys vorhanden.\n";
  close TEXT;
}
```

10. Schreiben Sie eine Subruotine my_reverse, die eine Zeile als Argument bekommt und die Elemente in umgekehrter Reihenfolge mit print ausgibt.
ACHTUNG: Es darf die Systemroutine reverse nicht verwendet werden, alles muss innerhalb der Subroutine mit split und Arrays programmiert werden.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: schreibt Subroutine, die Woerter in umgedrehter Reihenfolge
zurueckgibt

use strict;
use utf8;
use locale;
{
  my ($zeile, $zeile_rueckwaerts);

  print "Bitte geben Sie eine Textzeile ein >>> ";
  chomp($zeile = <>);

  &my_reverse($zeile);
}

sub my_reverse($) {
  my $zeile = $_[0];
  my $zeile_rueckwaerts = "";
  my @woerter = split(/ /,$zeile);
  for(my $i = (scalar(@woerter)-1); $i >= 0; $i--) {
    if($i==(scalar(@woerter)-1)) {
      $zeile_rueckwaerts = $woerter[$i];
    }
    else {
      $zeile_rueckwaerts = "$zeile_rueckwaerts $woerter[$i]";
    }
  }
  print "$zeile_rueckwaerts\n";
}
```

11. Schreiben Sie eine PERL- Subroutine print_words_with_index , das durch eine Liste von Wörter läuft und jedes Wort zusammen mit der Position innerhalb der Wortliste ausdruckt.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: Subroutine, die jedes Wort in Liste mit Position ausdruckt

use strict;
use utf8;
use locale;
{
  my @woerter = ("der", "mann", "und", "die", "frau");
  &print_words_with_index(@woerter);
}

sub print_words_with_index(@) {
  my @woerter = @_;
  for(my $i=1;$i<=scalar(@woerter);$i++) {
    print "Wort $i = $woerter[$i-1]\n";
  }
}
```