Musterlösung zu Blatt 6:

1. Wie lautet das PERL Programm, das mit Hilfe einer Schleife jede Zahl zwischen 0 und 30 ausgibt und bei jeder durch 3 teilbaren Zahl dazu schreibt: "Ich bin durch 3 teilbar".

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: durch 3 teilen
use strict;
{
  my $i;
  for ($i=0; $i<=30; $i++) {
    if ($i % 3 == 0 && $i!=0) {
      print "$i: Ich bin durch 3 teilbar.\n";
    } else {
      print "$i\n";
    }
  }
}
```

2. Erweitern Sie die letzte Aufgabe dahingehend, dass der Teiler 3 nicht mehr fest vorgegeben ist, sondern der Benutzer nach einem beliebigen Teiler gefragt wird. Mit diesem Teiler soll die Teilbarkeit mit dem Multiplikationsfaktor nach folgendem vorgebenen Muster ausgegeben werden:

```
Ausgabe:
Bitte geben Sie den Teiler ein >
Eingabe:
5
Ausgabe:
0,1,2,3,4
5 : ich bin durch 5 teilbar, der Multiplikator ist 1
6,7,8,9,
10 : ich bin durch 5 teilbar, der Multiplikator ist 2
…
```

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: durch 3 teilen
use strict;
{
  my $i;
  my $teiler;
  print "Bitte geben Sie den Teiler ein:>>>\n";
  chomp($teiler = <>);

  for ($i=0; $i<=30; $i++) {
    if ($i % $teiler == 0 && $i!=0) {
      print "\n$i : Ich bin durch $teiler teilbar, der Multiplikator ist ", $i/$teiler,
".\n";
    } else {
      print "$i,";
    }
  }
}
```

3. Schreiben Sie ein Programm, dem Sie eine Zeile eingeben können, und geben Sie nur die Vokale (aeiou) der Zeile auf dem Terminal aus.
Tipp: "Zeile einlesen, mit split zerlegen und durch das Array mit einer for Schleife laufen"

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: liest aus Zeile nur die Vokale aus
use strict;
{
  my $zeile;
  my (@buchstaben,@vokale);
  my $buchstaben;
  my $vokal;

  print"Bitte geben Sie eine Zeile ein:>>>\n";
  chomp($zeile=<>);
  $zeile=lc($zeile);
  @buchstaben=split(//,$zeile);
  @vokale=("a","e","i","o","u");

  foreach $buchstaben(@buchstaben){
    foreach $vokal(@vokale){
      if ($buchstaben eq $vokal){
        print"$buchstaben";
      }
    }
  }
  print "\n";
}
```

4. Holen Sie den Text eines Interviews von Gerhard Polt in der Süddeutschen Zeitung : "»Ich sinnlose vor mich hin... und das mit Begeisterung!« Gerhard Polt spricht über die Langeweile. Von Alex Rühle" mit folgendem Befehl:
lynx -dump http://sz-magazin.sueddeutsche.de/texte/anzeigen/36659
und sichern Sie im Verzeichnis aufg3 den Inhalt des Interviews in der Datei 'artikel.txt'.

```
cd aufg3
lynx -dump http://sz-magazin.sueddeutsche.de/texte/anzeigen/36659 >
artikel.txt
```

5. Schreiben Sie ein PERL Programm, das den Text aus der Datei 'artikel.txt' liest und splitten Sie die einzelnen Zeilen, damit Sie das erste Wort jeder Zeile auf dem Teminal ausgeben können.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: gibt erstes Wort der Zeile einer Datei aus
use strict;
{
  my $artikel = "artikel.txt";
  my $zeile;
  my @woerter;
  my $woerter;
  open(ARTIKEL,$artikel);

  while($zeile=<ARTIKEL>){
    chomp $zeile;
    @woerter=split(/ /,$zeile);
    #($woerter[0]) {                    # falls das Wort nicht leer ist
      print "$woerter[0]\n";
    #}
  }
  close (ARTIKEL);
}
```

Achtung: es ist auch in Ordnung wenn ihr leere Wörter mit ausgebt, es war hier noch nicht verlangt zu testen ob am Anfang der Zeile auch wirklich ein Wort steht.

6. Schreiben Sie ein PERL Programm, das den Text aus der Datei 'atrikel.txt' liest und eine einspaltige Wortliste des Textes erzeugt. Speichern Sie die Wortliste in der Datei 'wortliste.txt' .

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: erzeugt einspaltige Wortliste in wortliste.txt
use strict;
{
  my $artikel = "artikel.txt";
  my $output = "wortliste.txt";
  my ($zeile, @woerter, $woerter, $element);
  open(ARTIKEL,$artikel);
  open(WORTLISTE, ">$output");

  while($zeile=<ARTIKEL>){
    chomp $zeile;
    @woerter=split(/ /,$zeile);
```

```
    foreach $element (@woerter) {
      if ($element) {
        print "$element \n";
        print WORTLISTE "$element \n";
      }
    }
  }
  close (ARTIKEL);
  close (WORTLISTE);
}
```

7. Erweitern Sie das vorherige Programm dahingehend, das auch die Anzahl der Wörter, der Zeilen und die Anzahl der Zeichen (keine Spaces mitzählen!) in der Datei 'artikel.txt' gezählt werden.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: erzeugt einspalte Wortliste in wortliste.txt und zaehlt
use strict;
{
  my $artikel = "artikel.txt";
  my $output = "wortliste.txt";
  my ($zeile, @woerter, $woerter, $element);
  my $anzZeilen = 0;
  my $anzWoerter = 0;
  my $anzBuchstaben = 0;

  open(ARTIKEL,$artikel);
  open(WORTLISTE, ">$output");

  while($zeile=<ARTIKEL>){
    $anzZeilen++;
    chomp $zeile;
    @woerter=split(/ /,$zeile);

    foreach $element (@woerter) {
      if ($element) {
        $anzWoerter++;
        $anzBuchstaben = $anzBuchstaben + length($element);
        print "$element \n";
        print WORTLISTE "$element \n";
      }
    }
  }

  print "Ihr Text hat $anzZeilen Zeilen, $anzWoerter Woerter und
$anzBuchstaben Buchstaben!\n";

  close (ARTIKEL);
  close (WORTLISTE);
}
```

8. Vergleichen Sie das Ergebnis der letzten Aufgabe mit der Ausgabe des UNIX-Befehls wc

wc artikel.txt

9. Erweitern Sie das vorherige Programm dahingehend, dass Sie am Anfang des Progamms nach einem Wort gefragt werden und ausgeben ob das Wort, bzw. wie oft das Wort in der Datei 'artikel.txt' vorkommt.

```
#!/usr/bin/perl
# Autor: Nicola Greth
# Programm: sucht nach Wort in Wortliste
use strict;
{
  my $artikel = "artikel.txt";
  my $output = "wortliste.txt";
  my ($zeile, @woerter, $woerter, $element, $eingabe);
  my $anzZeilen = 0;
  my $anzWoerter = 0;
  my $anzBuchstaben = 0;
  my $anzEingabe = 0;
  open(ARTIKEL,$artikel);
  open(WORTLISTE, ">$output");

  print"Nach welchem Wort soll ich fuer Sie suchen? Eingabe:>>>\n";
  chomp($eingabe = <>);

  while($zeile=<ARTIKEL>){
    $anzZeilen++;
    chomp $zeile;
    @woerter=split(/ /,$zeile);

    foreach $element (@woerter) {
      if ($element) {
        $anzWoerter++;
        $anzBuchstaben = $anzBuchstaben + length($element);
        print "$element \n";
        print WORTLISTE "$element \n";
      }
        if ($element eq $eingabe) {
        $anzEingabe++;
      }
    }
  }

  print "Ihr Text hat $anzZeilen Zeilen, $anzWoerter Woerter und
$anzBuchstaben Buchstaben!\n";
  print "Ihr Wort $eingabe kommt im Text $anzEingabe mal vor!\n";
  close (ARTIKEL);
  close (WORTLISTE);
}
```

10. Vergleichen Sie das Ergebnis der letzten Aufgabe mit den Werten, die die Lösung der Aufgabe 1 aus dem letzten Aufgabenblatt liefert.

Wort	**Wert Übung 6**	**Wert Übung 5**
die	*44*	*45*
und	*33*	*36*
der	*34*	*36*
den	*16*	*22*
wie	*9*	*15*

11. Wie sind folgende Zeichenketten in UTF-8 - und in ISO-Latin-1 Kodierung abgespeichert? (Tipp: Speichern Sie die Wörter in einer UTF-8 und dann in einer ISO-Latin 1 Datei und betrachten Sie, analog zur Vorlesung, den octal Dump der Datei mit dem unix-Befehl od)

a) Zeichenkette : 'Anton'

od testUTF8
Ergebnis:
0000000 067101 067564 005156 000012
0000007

od testISOLATIN
Ergebnis:
0000000 067101 067564 005156 000012
0000007

b) Zeichenkette : 'Dez. 2010, 12:45'

od testUTF8
Ergebnis:
0000000 062504 027172 031040 030460 026060 030440 035062 032464
0000020 005012
0000022

od testISOLATIN
Ergebnis:
0000000 062504 027172 031040 030460 026060 030440 035062 032464
0000020 005012
0000022

c) Zeichenkette : 'äää üüü ööö'

od testUTF8
Ergebnis:
0000000 122303 122303 122303 141440 141674 141674 020274 133303
0000020 133303 133303 005012
0000026

od testISOLATIN
Ergebnis:
0000000 162344 020344 176374 020374 173366 005366 000012
0000015

Wo unterscheiden sich die Codierungen?

Die Codierungen unterscheiden sich bei den Umlauten. Umlaute werden in UTF-8 als Mehrbytezeichen dargestellt.