1. Verwenden Sie folgenden UNIX Befehl um aus der Datei artikel.txt der Aufgabe 3 eine Liste aller vorkommenden Wörter, zusammen mit Ihrer Frequenz zu erstellen. Tippen Sie folgenden UNIX Befehl ab und betrachten Sie das Resultat: cat artikel.txt | tr '[:upper:]' '[:lower:]' | tr -s ' ' '\n' | sort | uniq -c | sort -nr Was sind die 5 am häufigsten vorkommenden Wörter?

cat artikel.txt	**zeigt Datei an**
| tr '[:upper:]' '[:lower:]'	**ersetzt Großbuchstaben mit Kleinbuchstaben**
| tr -s ' ' '\n'	**ersetzt Spaces durch Newlines (-s mehrere Vorkommen durch eines ersetzen)**
| sort	**sortiert**
| uniq -c	**entfernt Duplikate (-c Anzahl der Vorkommen)**
| sort -nr	**sortiert numerisch rückwärts**

45 die, 36 und, 36 der, 22 den, 15 wie

2. Lesen Sie eine Textzeile vom Terminal ein und berechnen Sie die Anzahl der Vokale a,e,i,o,u. Geben Sie die Anzahl des jeweiligen Vokals aus.

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: Vokalzaehler

use strict;
{
  my ($zeile, $buchst, @buchst);
  my $a = 0;
  my $e = 0;
  my $i = 0;
  my $o = 0;
  my $u = 0;

  print "Bitte geben Sie eine Zeile ein: >>> \n";
  chomp($zeile = <>);

  @buchst = split(//,$zeile);

  foreach $buchst(@buchst){
    if (lc($buchst) eq "a"){
      $a++;
    }
    if (lc($buchst) eq "e"){
      $e++;
    }
    if (lc($buchst) eq "i"){
      $i++;
    }
    if (lc($buchst) eq "o"){
      $o++;
    }
```

```
    if (lc($buchst) eq "u"){
      $u++;
    }
  }


  print "Der Buchstabe a kommt $a mal vor.\n";
  print "Der Buchstabe e kommt $e mal vor.\n";
  print "Der Buchstabe i kommt $i mal vor.\n";
  print "Der Buchstabe o kommt $o mal vor.\n";
  print "Der Buchstabe u kommt $u mal vor.\n";
}
```

3. Wie lautet das PERL Programm, das zwischen 0 und 20 jede gerade Zahl ausgibt.

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: gibt jede gerade Zahl zwischen 0 und 20 aus

use strict;

{
  my $i;

  for ($i = 0; $i <= 20; $i=$i+2){
    print "$i\n";
  }
}
```

oder

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: gibt jede gerade Zahl zwischen 0 und 20 aus

use strict;

{
  my $i = 0;

  while ($i <= 20) {
    if ($i % 2 == 0) {
      print "$i\n";
    }
    $i++;
  }
}
```

4. Schreiben Sie ein PERL Programm das eine Zahl einliest und testet,
ob die Zahl eine gerade Zahl, ob sie gleich -6, -8 oder -10 ist und ob die Zahl eine negative Zahl ist, aber ja nicht kleiner als -50 ist.

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: testet Bedingungen fuer Zahlen

use strict;
{

  my $zahl;

  print "Bitte geben Sie eine Zahl ein:>>>\n";
 chomp( $zahl = <>);

  if ($zahl%2 == 0) {
    print "Ihre Zahl ist gerade.\n";
  }

  else {
    print "Ihre Zahl ist ungerade.\n";
  }

  if ($zahl == -6 || $zahl == -8 || $zahl == -10) {
    print "Ihre Zahl ist $zahl.\n";
  }

  else {
    print "Ihre Zahl ist weder -6, noch -8, noch -10.\n";
  }

  if ($zahl < 0) {
    print "Ihre Zahl ist negativ.\n";

    if ($zahl < -50) {
      print "Ihre Zahl ist kleiner als -50.\n";
    }

    else {
      print "Ihre Zahl ist nicht kleiner als -50.\n";
    }
  }

  else {
    print "Ihre Zahl ist eine positive Zahl.\n";
  }

}
```

5. Wie lautet das PERL Programm, das jede Zahl zwischen 0 und 30 ausgibt und immer wenn 5 Zahlen ausgegeben wurden, ausgibt: "das waren wieder fuenf Zahlen".

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: Gibt alle Zahlen bis 30 aus und meldet nach fuenf Zahlen

use strict;

{
  my $i;
  my $zaehler = 0;

  for($i = 0; $i <= 30; $i++){
    print "$i\n";
    $zaehler++;

    if ($zaehler == 5){
      print "Das waren wieder 5 Zahlen.\n";
      $zaehler = 0;
    }
  }
}
```

6. PASSWORT-Programm: Schreiben Sie ein PERL Programm, das den Benutzer maximal 5 Mal auffordert eine intern definierte Zeichenkette zu erraten.
Falls die Zeichenketten nicht identisch sind, soll eine Fehlermeldung ausgegeben werden und der Benutzer noch höchstens 4 Mal die Möglichkeit haben eine neue Zeichenkette einzugeben.

siehe Musterlösung Aufgabe 4

7. Lesen Sie ein Wort ein und schreiben Sie ein Programm, das entscheidet, ob der erste und letzte Buchstabe des Wortes identisch ist. (Tipp: Verwenden Sie die eingebauten Systemroutinen: length und split, siehe im Skript)

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: testet auf gleichen Anfangs- und Endbuchstaben

use strict;

{
  my ($word, $laenge, @buchst);

  print "Bitte geben Sie ein Wort ein: >>> \n";
  chomp($word = <>);

  $laenge = length($word);
  @buchst = split(//,$word);
```

```
  if (lc($buchst[0]) eq lc($buchst[$laenge-1])) {
    print "Anfangs- und Endbuchstabe sind identisch!\n";
  }

else {
    print "Anfangs- und Endbuchstabe sind nicht identisch!\n";
  }
}
```

8. Schreiben Sie ein PERL Programm palindrom.perl, das ein Wort einliest und entscheidet, ob das Wort ein Palindrom ist.

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: Palindromtester mit reverse

use strict;

{
  my ($word, $word_lower, $ word_reverse);

  print "Bitte geben Sie ein Wort ein:>>> ";
  chomp($word = <>);

  $word_lower = lc($word);
  $word_reverse = reverse($word_lower);

  if ($word_reverse eq $word_lower) {
    print "$word ist ein Palindrom.\n";
  }

  else {
    print "$word ist kein Palindrom.\n";
  }
}
```

oder ohne reverse:

```
#!usr/bin/perl
# Autor: Nicola Greth
# Programm: Palindromtester ohne reverse

use strict;
{
  my ($word, $word_lower,@buchst,$palindrom,$i);

  print "Bitte geben Sie ein Wort ein:>>> \n";

  chomp($word = <>);
  $word_lower = lc($word);
```

```
  @buchst = split(//, $word_lower);
  $palindrom = "true";

  for ($i = 0; $i < length($word)/2; $i++) {
    if ($buchst[$i] ne $buchst[length($word)-1-$i]) {
      $palindrom = "false";
    }
  }

  if ($palindrom eq "true") {
    print "$word ist ein Palindrom!\n";
  }

  else {
    print "$word ist kein Palindrom!\n";
  }
}
```

9. Geben Sie Ziffernfolge der folgenden Zahlen im Binär-/Hexamdezimal/Dezimal- und Oktalsystem an
a) Dezimalzahl 16

16 : 2 = 8 Rest 0
8 : 2 = 4 Rest 0
4 : 2 = 2 Rest 0
2 : 2 = 1 Rest 0
1 : 2 = 0 Rest 1
Binär: 10000

16 : 16 = 1 Rest 0
1 : 16 = 0 Rest 1
Hexadezimal: 10

16 : 8 = 2 Rest 0
2 : 8 = 0 Rest 2
Oktal: 20

b) Oktalzahl 777

$777 = 7*8^2 + 7*8^1 + 7*8^0 = 511$
Dezimal: 511

511 : 2 = 255 Rest 1
255 : 2 = 127 Rest 1
127 : 2 = 63 Rest 1
63 : 2 = 31 Rest 1
31 : 2 = 15 Rest 1
15 : 2 = 7 Rest 1
7 : 2 = 3 Rest 1
3 : 2 = 1 Rest 1
1 : 2 = 0 Rest 1
Binär: 111111111

511 : 16 = 31 Rest 15
31 : 16 = 1 Rest 15
1 : 16 = 0 Rest 1
Hexadezimal : 1FF

c) Hexadezimalzahl FF

$FF = 15*16^1 + 15*16^0 = 255$
Dezimal: 255

255 : 2 = 127 Rest 1
127 : 2 = 63 Rest 1
63 : 2 = 31 Rest 1
31 : 2 = 15 Rest 1
15 : 2 = 7 Rest 1
7 : 2 = 3 Rest 1
3 : 2 = 1 Rest 1
1 : 2 = 0 Rest 1
Binär: 11111111

255 : 8 = 31 Rest 7
31 : 8 = 3 Rest 7
3 : 8 = 0 Rest 3
Oktal: 377

d) Dezimalzahl 512

512 : 2 = 256 Rest 0
256 : 2 = 128 Rest 0
128 : 2 = 64 Rest 0
64 : 2 = 32 Rest 0
32 : 2 = 16 Rest 0
16 : 2 = 8 Rest 0
8 : 2 = 4 Rest 0
4 : 2 = 2 Rest 0
2 : 2 = 1 Rest 0
1 : 2 = 0 Rest 1
Binär: 1000000000

512 : 8 = 64 Rest 0
64 : 8 = 8 Rest 0
8 : 8 = 1 Rest 0
1 : 8 = 0 Rest 1
Oktal : 1000

512 : 16 = 32 Rest 0
32 : 16 = 2 Rest 0
2 : 16 = 0 Rest 2
Hexadezimal: 200