



In der Informatik gibt es eine Vielzahl von Sortieralgorithmen, einen einfachen wollen wir uns vorerst noch ansehen:

Der Bubble-Sort-Algorithmus

Wir haben nicht sortiertes Datenfeld, z.B.

j

17	5	3	7
----	---	---	---

Wir fangen bei $j[0]$ an und vergleichen ihn mit dem Feld daneben, also $j[1]$.
Wenn der linke Nachbar größer ist als der rechte so tauschen wir die beiden Werte aus!

Nun vergleichen wir das nächste Pärchen, also Position $j[1]$ mit $j[2]$.

Wenn wir das ganze Datenfeld abgearbeitet haben, können wir zumindest eines sicher sagen, die größte Zahl ist nach ganz oben gewandert.

Wir wiederholen diesen Vorgang solange bis alle Felder sortiert sind.



Aufgabe 1: Schreibe den Code zum Vertauschen von zwei Variablen, falls die linke Zahl größer ist als die rechte!

Lösungsansatz:

```
// es werden in einem Array $j das Feld mit dem Index $a mit seinem rechten Nachbarn vertauscht
```

```
if ($j[a]>$j[a+1])
{
    $Schublade=$j[$a];
    $j[$a]=$j[$a+1];
    $j[$a+1]=$Schublade;
}
```

Aufgabe 2: Packe eine Schleife um den Code von Aufgabe 1, so dass dieser bei \$a=0 anfängt und bis zum vorletzten Element des Arrays die Prüfung durchführt!

Lösungsansatz:

```
for ($i=0; $i<(count($j)-1); $i++)
{
    if ($j[a]>$j[a+1])
    {
        $Schublade=$j[$a];
        $j[$a]=$j[$a+1];
        $j[$a+1]=$Schublade;
    }
}
```



Aufgabe 3: Packe eine Schleife um den Code von Aufgabe 2, so dass dieser den gesamten Vorgang so viele Male durchführt, wie das Array Elemente hat!

Lösungsansatz:

```
for ($i1=0; $i1<(count($j); $i1++)
{
  for ($a=0; $a<(count($j)-1); $a++)
  {
    if ($j[$a]>$j[$a+1])
    {
      $Schublade=$j[$a];
      $j[$a]=$j[$a+1];
      $j[$a+1]=$Schublade;
    }
  }
}
```



Damit ist die Aufgabe auch schon fertig, aber noch nicht gut!

Schreibe eine passende Funktion `BubbleSort($j)`, die einen unsortierten Array übergibt und einen sortierten Array zurückliefert! Kommentiere den Code

```
function BubbleSort($j){

// äußere Schleife, bei BubbleSort wandert bei jedem Durchgang die größte Zahl nach rechts
// also wird der innere Teil für jedes Element des Arrays einmal aufgerufen
for ($i1=0; $i1<count($j); $i1++)
{

// innere Schleife, bei BubbleSort wird jeweils das größte Element nach rechts geschoben, dazu
// wird das ganze Feld von vorne nach hinten abgearbeitet
for ($i=0; $i<(count($j)-1); $i++)
{

// es werden immer Paare verglichen, ist der linke Wert größer als der rechte
// werden die Zahlen im Array vertauscht

    if ($j[$i]>$j[$i+1])
    {

// hier werden die Zahlen eines Arrays vertauscht
// dazu benötigt man die Hilfsvariable $Schublade zum Merken des Wertes
        $Schublade=$j[$i];
        $j[$i]=$j[$i+1];
        $j[$i+1]=$Schublade;
    }
}
}
return $j;
}
```



Die Funktion muss mit `$zahl=BubbleSort($j)` aufgerufen werden, da Variablen außerhalb einer Funktion die innerhalb nicht kennen!

Das Programm ist jetzt schon sehr schön, es kann aber noch verbessert werden:

a) Es muss bei den Vergleichen nicht immer bis zum Ende verglichen werden, da bei jedem Durchgang (innere Schleife) ein Feld rechts hinzukommt, das schon sortiert ist.

b) Es kann bei jedem Durchgang der inneren Schleife ein Flag gesetzt werden, ob bei diesem Durchgang nochmals vertauscht werden musste. Falls nicht, kann der Vorgang nämlich sofort beendet werden, da das Feld schon sortiert ist. Z.B. wenn die Liste schon fast sortiert ist.

```
<?php
for ($i1=0; $i1<count($j); $i1++)
{
    $flag1=false;
    for ($a=0; $a<(count($j)-(1+$i1)); $a++)
    {
        if ($j[$a]>$j[$a+1])
        {
            $Schublade=$j[$a];
            $j[$a]=$j[$a+1];
            $j[$a+1]=$Schublade;
            $flag1=true;
        }
    }
    if (flag1==false) $i1=count($j);
}
for ($i=0;$i<count($j);$i++){
echo $j[$i], "<br>";
}
?>
```