

a)

Algorithmus:

Speichere den Wert des ersten Elements in einer Variablen.

Setze dann das erste Element auf den Wert des letzten Elements,

und dann das letzte Element auf den Wert der Variablen.

```
public void erstesLetztes()  
{  
    int temp;  
    temp = liste[0];  
    liste[0] = liste[liste.length - 1];  
    liste[liste.length - 1] = temp;  
}
```

b)

Algorithmus:

Wiederhole vom Anfang bis zur Mitte des Arrays, mit i als Index:

Vertausche das i-te Element mit dem i-t letzten Element

Zum Vertauschen nutze eine Variable (wie bei Aufg. a)

Hinweis: bei ungerader Anzahl wird das mittlere Element nicht berührt.

```
public void reihenfolgeUmdrehen()  
{  
    int i, temp;  
    for (i = 0; i < liste.length / 2; i++)  
    {  
        temp = liste[i];  
        liste[i] = liste[liste.length - i - 1];  
        liste[liste.length - i - 1] = temp;  
    }  
}
```

c)

Algorithmus:

Speichere das erste Element in einer Variablen.

Wiederhole vom ersten bis zum vorletzten Element:

Setze das aktuelle Element auf den Wert seines Nachfolgers

Nach der Schleife setze das letzte Element auf den Wert der Variablen.

```
public void schiebeLinks()  
{  
    int i, temp;  
    temp = liste[0];  
    for (i = 0; i < liste.length - 1; i++)  
    {  
        liste[i] = liste[i+1];  
    }  
    liste[liste.length - 1] = temp;  
}
```

d)

Algorithmus:

Speichere das letzte Element in einer Variablen.

Wiederhole vom letzten bis zum zweiten Element (rückwärts):

Setze das aktuelle Element auf den Wert seines Vorgängers

Nach der Schleife setze das erste Element auf den Wert der Variablen.

```
public void schiebeRechts()
{
    int i, temp;
    temp = liste[list.length-1];
    for (i = liste.length-1; i > 0; i--)
    {
        liste[i] = liste[i-1];
    }
    liste[0] = temp;
}
```

e)

Algorithmus:

Wiederhole 50 mal:

Setze zwei Variablen a und b jeweils auf eine zufällige Zahl zwischen 0 und Länge – 1

Tausche die Elemente an Position a und b (mithilfe einer Variablen)

```
public void mischen()
{
    int i, a, b, temp;

    for (i = 0; i < 50; i++)
    {
        a = (int) (Math.random() * liste.length);
        b = (int) (Math.random() * liste.length);
        temp = liste[a];
        liste[a] = liste[b];
        liste[b] = temp;
    }
}
```

e)

Algorithmus:

Eine Variable „anzahl“ zählt, wie viele negative Zahlen das Array enthält.

Setze sie zu Beginn auf 0.

Eine zweite Variable i steht für den aktuellen Index, setze sie auch auf 0.

Wiederhole, solange i + anzahl kleiner als die Anzahl der Elemente ist:

Solange das Element an Position i + anzahl negativ ist (und das Ende nicht erreicht ist):

erhöhe anzahl um 1

Falls eine negative Zahl gefunden wurde (anzahl &gt; 0)

und die letzte gefundene negative Zahl nicht am Ende des Array steht:

Setze das aktuelle Element (an Pos. i) auf die Zahl hinter der letzten negativen

Erhöhe i um 1

Jetzt sind alle nicht-negativen Zahlen nach vorn geschoben.

Die letzten „anzahl“ Elemente müssen mit 0 überschrieben werden.

```
public void negativeLoeschen()
{
    int i, anzahl;
    anzahl = 0;

    for (i = 0; i < liste.length - anzahl; i++)
    {
        while (i + anzahl < liste.length && liste[i + anzahl] < 0)
        {
            anzahl++;
        }
        // Sonderfall: falls am Ende eine negative Zahl steht,
        // wird nichts vorgezogen
        if (anzahl > 0 && i + anzahl < liste.length)
        {
            liste[i] = liste[i+anzahl];
        }
    }

    for (i = liste.length-anzahl; i < liste.length; i++)
    {
        liste[i] = 0;
    }
}
```