

Aufgabe 1: (6 + 2)

Entwerfen Sie eine Methode `ausgabe(...)`, die ein gegebenes und bereits gefülltes Array vom Typ `Ganzzahl` so ausgibt, das nur jedes 2. Element ausgegeben wird.

Entwerfen Sie die Methode in allgemeiner Form so, dass der zweite Parameter bestimmt welche Elemente ausgegeben werden. 1 → jedes Element, 2 → jedes 2. Element usw.

Hier der bereits existierende Quellcode:

```
package ka_arrays_gruppe_b;

import AlgoTools.IO;

public class NurZweitesElementAusgeben {

    public static void main(String[] args) {

        int[] liste_a = fuelleArray(8);

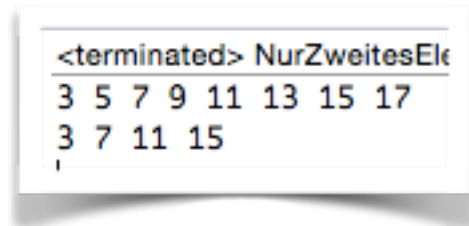
        ausgabe(liste_a,1);

        ausgabe(liste_a,2);

    }

}
```

Hier die entsprechende Ausgabe:



```
<terminated> NurZweitesEle
3 5 7 9 11 13 15 17
3 7 11 15
,
```

Hier wäre Platz für Ihre Lösung

```
public static void ausgabe(int[] l, int n) {

    for (int i = 0; i < l.length; i++) {

        if(i%n==0) {

            IO.print(l[i] + " ");

        }

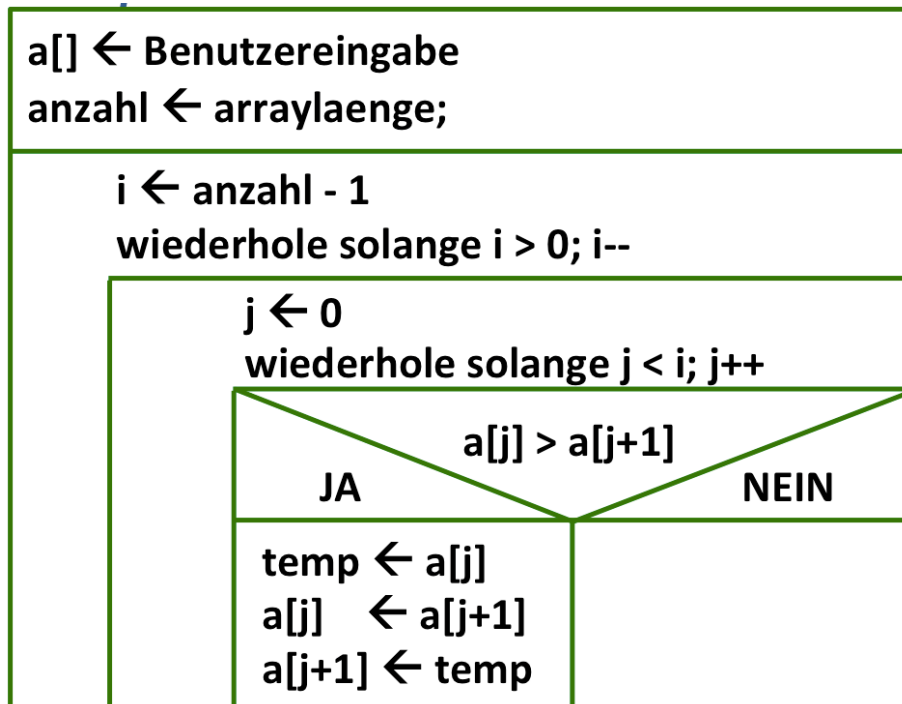
    }

    IO.println();

}
```

Aufgabe 2: (6 + 3)

Es gibt mehrere Möglichkeiten Werte innerhalb eines Arrays zu sortieren. Hier sehen Sie das Struktogramm des Prinzips Bubble Sort.



- Implementieren Sie diesen Algorithmus anhand des gegebenen Struktogramms.
- Erklären Sie ob der Algorithmus aufsteigend oder absteigend sortiert.

Lösungsvorschlag:

```

int n = IO.readInt("Wie viele Elemente ?: ");

int[] a = einlesen(n);

int anzahl = a.length;

for(int i=anzahl-1; i>0; i--)
{
    for(int j = 0; j<i; j++)
    {
        if( a[j] > a[j+1] )
        {
            int temp = a[j];
            a[j]      = a[j+1];
            a[j+1]    = temp;
        }
    }
}

```

Aufgabe 3: (8)

Begründen Sie zu folgendem Quellcode die Ausgabe:

```
public static void main(String[] args) {

    int anzahl = 8, i = 0, j = 0, k = 0;
    int[] a = new int[anzahl / 2];
    int[] b = new int[anzahl / 2];

    while (k++ < anzahl) {

        if (k % 2 == 1) {

            a[i++] = k;

        } else {

            b[j++] = k;

        }

    }

    for (int m = 0; m < b.length; m++) {
        IO.print((a[m] + b[b.length - 1 - m]) + " ");
    }

}
```

Lösungsvorschlag:

- Die beiden Arrays besitzen jeweils eine Länge von 4 Stellen
- Das Array a erhält alle ungeraden, das Array b alle geraden Werte
- $a \leftarrow \{1, 3, 5, 7\}$
- $b \leftarrow \{2, 4, 6, 8\}$
- Jetzt wird jeweils das erste und das letzte Element aufaddiert, danach das zweite und das vorletzte

$1 + 8, 3 + 6, 5 + 4, 7 + 2$

Die Ausgabe lautet: **9 9 9 9**

Aufgabe 4: (8)

Ein Array kann maximal 10 Werte vom Typ Ganzzahl aufnehmen. Solange der Benutzer die Werte in einer aufsteigenden Reihenfolge eingibt ist alles in Ordnung.

Sobald aber ein Wert eingegeben wird der nicht in diese bereits sortierte Eingabe hineinpasst wird die Eingabe abgebrochen.

Danach werden die bereits eingegeben sortierten Werte einschließlich deren Anzahl ausgegeben.

Geben Sie den dafür benötigten Quellcode an.

Lösungsvorschlag folgt auf der nächsten Seite.

Lösungsvorschlag:

```
public static void main(String[] args) {

    int[] sortiereListe = new int[10];

    // Beachten Sie, dass ein neu erzeugtes Array mit 0ern
    // belegt ist.

    int anzahl = wieVieleWerteWurdenEingelesen(sortiereListe);

    // Wir uebergeben der Methode ausgabe die Information wie
    // viele Elemente ausgegeben werden.

    ausgabe(sortiereListe, anzahl);

}

public static int wieVieleWerteWurdenEingelesen(int [] l) {

    // Der erste Wert wird immer eingelesen

    int anzahl = 1;

    l[0] = IO.readInt("1. Wert: ");

    // Ab jetzt wird geprueft ob der Wert abgelegt wird oder nicht

    for (int i = 1; i < l.length; i++) {

        int eingabe = IO.readInt((i + 1) + ". Wert: ");

        if( eingabe > l[i-1]){

            // Wenn die Eingabe groesser gleich ist als das
            // Vorgaengerelement wird der Wert abgelegt.
            // und die anzahl erhoeht.

            l[i] = eingabe;
            anzahl++;

        }
        else{

            // Ansonsten verlassen wir die Schleife.

            break;

        }

    }

    return anzahl;

}

public static void ausgabe(int[] l, int n) {

    for (int i = 0; i < n; i++) {

        IO.print(l[i] + " ");

    }

    IO.println();

}
```