

**Aufgabe 1: "Reihen"****[4+4+4 Punkte]**

Gesucht sind Variationen des unten stehenden JAVA-Programms, welche jeweils die Ausgabe der unten angegebenen Reihen ermöglichen.

- a)     1 2 0 3 -1 4 -2 5 -3 6 -4 ...     (END = 10)  
 b)     1 2 3 5 8 13 21 34 ...     (END = 100)  
 c)     1 2 2 4 8 32 256 8192 ...     (END = 8192)

Sie können von dem hier angegebenen Code-Beispiel ausgehen. Ändern Sie nach Notwendigkeit die gelb markierten Code-Teile ab, um die jeweiligen Reihen zu generieren.

Setzen sie **END** auf die jeweils für die Reihe angegebenen Werte, um ihr jeweiliges Programm auszuprobieren.

Der dem Programm zugrundeliegende Algorithmus soll theoretisch Reihen bis zu beliebigen Werten von **END** generieren können.

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

        final long END = 200;    // max. REIHENWERT

        Out.print("\nVorgabe:    1 3 5 7 9 11 13 ...");
        Out.print("\nGeneriert: ");

        long val;    // REIHENWERT, der jeweils ausgegeben werden soll

        // TODO - AUSGABE DER REIHE -----
        for (val = 1; val <= END; val+=2) {

            Out.print(" " + val);    // AUSGABE DES WERTES

        }

        // -----

    } // end main()
} // end class Reihe
```

**Hinweise:**

- Überlegen Sie im Besonderen, welchen Startwert Sie für **val** und ggf. andere Variablen setzen müssen.
- Überlegen Sie, wie sich der Wert für **val** gegenüber dem vorangegangenen Wert/den vorangegangenen Werten ändert.
- Im oben gezeigten Beispiel ist der Wert von **val** gegenüber dem Vorgängerwert immer um 2 erhöht.
- Beispielsweise, für die Reihe **a)** ist die Veränderung: +1 -2 +3 -4 ....

**Abgabe:**

➔ 3 eigenstehende **JAVA-Programm**, welche jeweils eine der oben angegebenen Reihen ausgibt.

**Aufgabe 2: "Prüfziffernberechnung"****[12 Punkte]**

Die **Internationale Standardbuchnummer** (englisch: International Standard Book Number, **ISBN**) ist eine Nummer zur eindeutigen Kennzeichnung von Büchern und anderen selbständigen Veröffentlichungen mit redaktionellem Anteil, wie beispielsweise Multimedia-Produkten und Software. Es existieren dazu verschiedene Varianten, e.g. ISBN-10 und ISBN-13. Hier wollen wir uns auf ISBN-10 beschränken.

Zu Sicherungs- und Kontrollzwecken von ISBNs wird die **letzte Stelle als Prüfziffer** verwendet. Diese Prüfziffer der ISBN wird nach einfachen Regeln aus den übrigen Ziffern berechnet.

Man multipliziere die erste Ziffer mit eins, die zweite mit zwei, die dritte mit drei und so fort bis zur neunten Ziffer, die mit neun multipliziert wird. Man addiere die Produkte und teile die Summe ganzzahlig mit Rest durch 11. Der Divisionsrest ist die Prüfziffer. Falls der Rest 10 beträgt, ist die Prüf-"ziffer" ein "X".

**Beispiel einer ISBN-Prüfziffer:**

ISBN 3898641171 ("Sprechen Sie Java?")

$$3*1 + 8*2 + 9*3 + 8*4 + 6*5 + 4*6 + 1*7 + 1*8 + 7*9 =$$
$$3 + 16 + 27 + 32 + 30 + 24 + 7 + 8 + 63 = 210$$

$210 / 11 = 19$  Rest: 1 => **Prüfziffer korrekt!**

Schreiben Sie ein Programm, das eine **ISBN einliest und prüft, ob diese formal korrekt ist**, d.h. ob die Prüfziffernberechnung korrekt ist.

**Beispiele:**

ISBN: 3898641171  
korrekt!

ISBN: 389864117X  
nicht korrekt!

ISBN: 12345678901  
nicht korrekt!

ISBN: 1234X67890  
nicht korrekt!

ISBN: 123456789X  
korrekt!

➔ **JAVA-Programm**

**Hinweise:**

- Verwenden Sie zum Einlesen der ISBN die Methode `In.read()`.