

Wird den Bitmustern, die man für $b=31$ erhält noch ein weiteres Bit „0“ vorangestellt, so ergibt sich gerade die Darstellung für eine der 2^{31} in Java durch `int` darstellbaren nicht negativen Zahlen. Wenn Bitmuster in der Tabelle $b=31$ dann von $n=0, 1, \dots, 2^{31}-1$ durchnummeriert, listet diese für jedes n den zugehörigen Gray-Code auf, der wiederum mit vorangestelltem „0“-Bit als `int` in Java interpretiert werden kann. Es ist also implizit eine Funktion gegeben, die jeder positiven Integer in Java eine (in der Regel andere) positive Integer zuordnet.

Schreiben Sie eine Klasse `Gray.java`, die:

a.) [3 Punkte]:

Eine Funktion `public static int toGray(int n)` implementiert, die für nicht negative n den entsprechenden Gray-Code als nicht negative Integer liefert.

b.) [3 Punkte]:

Eine Funktion `public static int fromGray(int g)` implementiert, die für nicht negative Integer g , die einen Gray-Code darstellen sollen, den ursprünglichen Wert n , also gerade die Umkehrfunktion berechnet.

Hinweise:

- Die beiden Funktionen müssen im Wertebereich `{0, ..., Integer.MAX_VALUE}` korrekt funktionieren.
- Die Funktionen müssen `public static` sein, damit sie auch von anderen Java-Klassen (z.B. der bereitgestellten Testklasse `TestGray.java`) verwendet werden können.
- Die Abbildung ist bijektiv.
- **Testfälle:** `TestGray.java` (`javac *Gray.java && java TestGray`)
 - 1) `toGray(0) == 0?`
 - 2) `fromGray(0) == 0?`
 - 3) `toGray(n)^toGray(n+1)`
liefert einen Bit-Count von „1“ (Hamming-Abstand=1)
 - 4) `fromGray(toGray(n)) = n`

Abgabedatei im Goya: `Gray.java`

Abgabehinweise

- Die angegebenen Ausgaben zeigen nur ausgewählte Beispiele. Ihre Programme sollen für den gesamten Wertebereich korrekt funktionieren.
- Referenzrechner (dort sollte Ihre Lösung funktionieren):
star.informatik.hu-berlin.de
- Die **Abgabe** der Übungsaufgaben erfolgt **in Gruppen**, ideale Gruppengröße=2.
Wichtig: Erst Gruppe bilden, dann Lösungen hochladen!
- Folgende Datei ist **rechtzeitig über Goya** als **UTF-8** codierter Lösungs Quelltext abzugeben: `Gray.java`