

Dualzahlen:

Entwickeln Sie ein Programm, das für eine eingegebene Dezimalzahl (nur ganze Zahlen) die entsprechende Dualzahl berechnet. Lesen Sie eine Zahl von der Eingabeaufforderung ein. Dividieren Sie die Zahl sukzessive durch 2, solange, bis kein Rest mehr bleibt. Die Reste ergeben die Entsprechende Dualzahl.

zb.

$$100 / 2 = 50 \text{ R: } 0$$

$$50 / 2 = 25 \text{ R: } 0$$

$$25 / 2 = 12 \text{ R: } 1$$

$$12 / 2 = 6 \text{ R: } 0$$

$$6 / 2 = 3 \text{ R: } 0$$

$$3 / 2 = 1 \text{ R: } 1$$

$$1 / 2 = 0 \text{ R: } 1$$

Binärzahl : 1100100

Probe:

$$1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = 100$$

$$64 + 32 + 4 = 100$$

Gliedern Sie die Berechnungsmethode in eine Funktion aus, die eine int zahl übernimmt und eine Binärzahl (in Form eines Strings) zurückgibt.

Bonusaufgaben:

- Schreiben Sie eine weitere Funktion zur Umrechnung von Dezimalzahlen in Oktalzahlen. Gehen Sie dazu gleich vor wie bei der Konvertierung in Dualzahlen. Ersetzen Sie nur den Schritt "Divison durch 2" durch den Schritt "Divison durch 8".

$$100 / 8 = 12 \text{ R: } 4$$

$$12 / 8 = 1 \text{ R: } 4$$

$$1 / 8 = 0 \text{ R: } 1$$

$$1 * 8^2 + 4 * 8^1 + 4 * 8^0 = 100$$

$$64 + 32 + 4 = 100$$

- Generalisieren Sie ihre Funktion und ermöglichen Sie außerdem die Berechnung von Hexadezimalzahlen
- Achtung ! in Visual Basic wird eine ganzzahlige Division mit dem „\“ (Backslash) – Operator durchgeführt.