# Programmieren C: Bitoperationen: *n*-te Hexziffer eines `int`


## *Klaus Kusche*


Gesucht ist eine **Funktion** **`hexdigit`**. Diese Funktion hat *zwei Parameter*:
Eine *beliebige Zahl* **`z`** und einen *Index* **`n`**. Beides sind *vorzeichenlose* ganze Zahlen,
*deklariere* dir dafür **`uint`** als *Abkürzung* für den **Typ** **`unsigned int`**.

Als **`int`**-*Returnwert* soll die Funktion die **`n`**-*te Hex-Ziffer* liefern (also eine *Zahl* von **`0`** bis **`15`**),
wenn man **`z`** *in Hex-Darstellung* schreibt. Die *Nummerierung* der Hex-Ziffern beginnt *links*
(bei der höchstwertigen Stelle) mit **`1`**, wobei *führende Hex-Nullen nicht mitgezählt* werden.
Ist **`n`** gleich **`0`** oder *größer* als die Anzahl der Hex-Ziffern, soll die Funktion **`-1`** liefern.

**Beispiel:** Für **`z`** gleich **`0x00b0affe`** und **`n`** gleich **`3`** liefert die Funktion den Wert dezimal **`10`**
(für die Hex-Ziffer **`a`**): Das **`n`** beginnt beim **`b`** zu zählen, die *dritte Hex-Ziffer von links*
ist daher das **`a`**, und **`a`** entspricht dezimal **`10`**.

Mir ist jede richtige Lösung recht, aber empfehle folgendes Vorgehen:

- Ermittle mit einer *Schleife*, *wie viele Hex-Ziffern* die Hex-Darstellung von **`z`** *lang* ist,
  und zwar *ohne* führende Nullen.
  Das geht am leichtesten, indem man von einer *Kopie* von **`z`** *immer wieder*
  *die letzte Hex-Ziffer wegstreicht* (und dabei *mitzählt*), bis nur mehr **`0`** übrig ist.
- Fange dann die *Sonderfälle* von **`n`** ab.
- *Berechne* aus der gezählten Anzahl der Hex-Ziffern und dem **`n`**,
  *um wie viele Bits* man **`z`** verschieben muss,
  damit die *gesuchte Hex-Ziffer ganz hinten* steht, und sorge dafür,
  dass das Funktions-Ergebnis *nur aus dieser einen verschobenen Hex-Ziffer* besteht.

Die Funktion soll ***ohne* Multiplikation, Division oder Restrechnung** auskommen
(auch bei der Berechung der Anzahl der Schiebe-Bits!) und ***nicht* rekursiv** arbeiten.

Schreib dazu ein *einfaches Hauptprogramm zum Testen*
(**`z`** und **`n`** von der Befehlszeile einlesen, Ergebnis der Funktion ausgeben).

**Tipp:** Wenn du **`strtoul(`***`str`***`, NULL, 0)`** statt **`atoi(`***`str`***`)`** für **`z`** verwendest,
dann kannst du die Zahl auf der Befehlszeile wahlweise *dezimal* (wie üblich)
oder *hexadezimal* (mit **`0x`** am Anfang) oder *oktal* (mit **`0`** am Anfang) eingeben.