

Programmieren 1 Übung: Arrays: Lottoziehung

Klaus Kusche

Schreib ein Programm, das eine **Lotto-Ziehung** macht, d.h. 6 zufällige Zahlen zwischen 1 und 49 ausgibt.

Dazu reicht es aber nicht, einfach 6 Mal eine Zufallszahl zwischen 1 und 49 auszugeben:

- Erstens dürfen beim Lotto ja keine doppelten Zahlen vorkommen.
- Zweitens sollen die Zahlen schön in aufsteigender Reihenfolge angezeigt werden.

Verwende daher folgende Idee:

- Es wird ein Array mit 49 **bool**-Elementen angelegt.
- Der Wert eines jeden Elementes kann daher entweder **true** oder **false** sein: Wenn das Element Nummer **i** gleich **true** ist, so heißt das, dass die Zahl **i+1** gezogen wurde, ist es **false**, so wurde **i+1** nicht gezogen (+1, weil der Array-Index ja von 0 bis 48, die Lottozahlen aber von 1 bis 49 gehen). Nach einer Ziehung muss das Array daher sechs Mal true und sonst lauter false enthalten.

Dein Programm soll folgendes tun:

- Als Erstes werden alle Elemente des Arrays der Reihe nach auf **false** gesetzt.
- Dann wiederholt man 6 Mal Folgendes:
 - Man berechnet so lange immer wieder eine Zufallszahl z zwischen 0 und 48, bis man ein **z** gefunden hat, das bisher noch nicht gezogen wurde, d.h. für das entsprechende Array-Element Nummer **z** noch false ist.
 - Man setzt das Array-Element Nummer **z** auf **true**.
- Am Ende geht man noch einmal alle Elemente des Arrays der Reihe nach durch: Wenn ein Element den Wert **true** enthält, gibt man die entsprechende Zahl aus.

Achtung: Unsere Array-Elemente und Zufallszahlen gehen von 0 bis 48, die Lotto-Zahlen von 1 bis 49!

Hinweise:

- Das Programm erwartet keine Eingabe und keine Angaben auf der Befehlszeile.
- Mach oben im Programm schöne Konstanten für die Anzahl der Zahlen (49) und für die Anzahl der zu ziehenden Kugeln (6) !
- Achte darauf, dass dein Programm bei jedem Aufruf andere Zahlen zieht! Was musst du dafür am Anfang des Programms tun?
- Vergiss nicht auf **stdbool.h** .
Vermeide Prüfungen wie (... == **true**) : Wie prüft man einen **bool**-Wert?