

Lernziele:

Vertieft werden sollen die Kenntnisse über Schleifen und Arrays.

Aufgabe 20:

Schreiben Sie ein C-Programm, das den Mittelwert aus einer Folge von maximal 10 Zahlen berechnet. Die Zahlen sollen mittels `scanf` in ein Array eingelesen werden.

Der Mittelwert aus n Zahlen a_1, \dots, a_n ist definiert als:

$$avg(a_1, \dots, a_n) := \frac{a_1 + a_2 + \dots + a_n}{n}$$

Aufgabe 21:

Schreiben Sie ein C-Programm, das eine Integer-Zahl zu einer beliebigen Basis (2 bis 36) umwandelt. Die Zahl und die Basis sollen mit `scanf` eingelesen werden.

Aufgabe 22:

Schreiben Sie ein C-Programm, das aus einer Liste von Zahlen all diejenigen Zahlen ausgibt, die kleiner als ein Wert `max` sind.

Aufgabe 23:

Schreiben Sie ein C-Programm, das aus einer Liste von Zahlen all diejenigen Zahlen ausgibt, die mit 7 enden.

Aufgabe 24:

Schreiben Sie ein C-Programm, das aus einer Liste von Zahlen das kleinste Element bestimmt und ausgibt.

Ändern Sie Ihr Programm so, dass auch das Maximum ausgegeben wird.

Aufgabe 25:

Schreiben Sie ein C-Programm, das aus einer Liste von Zahlen das dichteste Zahlenpaar, also das Zahlenpaar mit dem geringsten Abstand zueinander, ausgibt.

Beispiel: von den Zahlen 42, 8, 15, 37, 19, 81, 64, 44, 9 haben die Zahlen 8 und 9 den geringsten Abstand.

Aufgabe 26:

Schreiben Sie ein C-Programm, das zu mehreren Geldbeträgen eine Münzliste erstellt, also eine Tabelle, in der angegeben ist, wieviel Stück der einzelnen Geldwerte man benötigt, um alle Geldbeträge ohne Wechselgeld auszahlen zu können.

Beispiel: 5432, 345 und 756

Betrag	500	200	100	50	20	10	5	2	1
5432	10	2			1	1		1	
345			1	1		2		1	
756	1	2		1			1		1
Summe	11	5	1	1	3	1	2	1	1

Aufgabe 27:

Schreiben Sie ein C-Programm, das das Pascalsche Dreieck ausgibt, d.h. die Binomialkoeffizienten sollen tabellarisch ausgegeben werden. Die Binomialkoeffizienten sind wie folgt definiert:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad \text{mit} \quad \binom{n}{n} = \binom{n}{0} = 1$$

Mögliche Ausgabe des Programms:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

Aufgabe 28: (Sieb des Eratosthenes)

Schreiben Sie ein C-Programm, das zu einer gegebenen ganzen Zahl $x > 0$ alle Primzahlen p mit $p < x$ findet. Verwenden Sie dazu das Sieb des Eratosthenes, das auf folgender Idee beruht:

- schreibe alle Zahlen von 2 bis x hintereinander auf
- solange noch Zahlen in der Liste stehen:
 - die erste Zahl der Liste ist eine Primzahl
 - streiche die erste Zahl und alle Vielfachen dieser Zahl aus der Liste