

Algorithmen und Datenstrukturen

Aufgabe 1 (AGS 3.1.6 und AGS 3.1.2 b, c)

Schreiben Sie jeweils ein Programm, das

- (a) zwei Zahlen einliest und das Maximum beider Zahlen ausgibt,
- (b) für eine Eingabe n die Fakultät $n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1$, berechnet und das Ergebnis ausgibt,
- (c) für eine Eingabe n eine Multiplikationstabelle für die Zahlen von 1 bis n ausgibt, und
- (d) die Primzahlen von 1 bis 1000 berechnet und ausgibt.

Aufgabe 2 (AGS 3.1.17)

Schreiben Sie ein Programm, das die Nutzer*innen eine zufällig bestimmte Zahl von 1 bis n raten lässt, wobei n eine Eingabe ist. Dabei soll so oft geraten werden, bis eine geratene Zahl mit der zufälligen Zahl übereinstimmen. Nach jedem Versuch soll nur ausgegeben werden, ob die geratene Zahl größer, kleiner, oder gleich der gesuchten Zahl ist. Was ist die optimale Ratestrategie?

Zusatzaufgabe 1 (AGS 3.2.47 a *)

In einem Spiel wird zunächst eine positive Zahl von Streichhölzern (Englisch: *matches*) festgelegt. Anschließend ziehen zwei Spieler abwechselnd jeweils 1, 2 oder 3 Streichhölzer, vorausgesetzt es sind noch entsprechend viele Streichhölzer übrig. Der Spieler, der das letzte Streichholz zieht, verliert das Spiel.

Vervollständigen sie das folgende C-Programm anhand der Kommentare, bei dem ein Computer das Streichholzspiel gegen einen Menschen spielt, so dass gilt:

- Der Mensch legt die initiale Zahl der Streichhölzer (*matches*) durch eine Eingabe fest. Der Computer führt den ersten Zug (*turn*) aus.
- Der Computer führt stets von Ihnen festzulegende, zulässige (aber ggf. suboptimale) Züge aus.
- Für jeden Zug des Menschen soll eine Zahl eingelesen werden, die festlegt wieviele Streichhölzer gezogen werden. Gehen Sie davon aus, dass alle Eingaben zulässig sind.

Geben Sie für eine Belegung für jede der Variablen A – F im folgenden C-Programm an!

```
1  #include <stdio.h>                12      // verbleibenden Streichhoelzer
2                                     13      D
3  int main(){                       14      } else {
4      int matches, turn;            15      // Zug des Menschen
5      scanf("%d", &matches);        16      E
6      turn = 1;                    17      }
7      while (A){                   18      }
8          if (turn == B){           19      // Bekanntgabe des Gewinners
9              // Zug des Computers  20      F
10             C
11             // Bekanntgabe der    21      return 0;
12                                     22      }
```

Zusatzaufgabe 2 (AGS 3.1.5 ★)

Für $n, k \in \mathbb{N}$ mit $k \leq n$ ist der Binomialkoeffizient $\binom{n}{k}$ definiert durch:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Schreiben Sie ein C-Programm, das den Binomialkoeffizienten für zwei eingegebene Zahlen n und k berechnet.