

Algorithmen und Datenstrukturen

Beachten Sie die Verlegung der Präsenzübungen zum Buß- und Bettag:

- die Übung Mi, 1. DS wird auf Dienstag, den 17. November in der 3. DS (APB/E023),
- die Übung Mi, 5. DS wird auf Dienstag, den 17. November in der 2. DS (APB/E001) verlegt.

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

Schreiben Sie jeweils ein Programm, das

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

Aufgabe 2 (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>           10      C
2                               11      // Bekanntgabe der
3  int main(){                 12      // verbleibenden Streichhoelzer
4      int matches, turn;      13      D
5      scanf("%d", &matches);  14      } else {
6      turn = 1;               15      // Zug des Menschen
7      while (A){              16      E
8          if (turn == B){     17      }
9          // Zug des Computers 18      }
```

```

19 // Bekanntgabe des Gewinners          21 return 0;
20 F                                     22 }

```

Zusatzaufgabe 1 (AGS 2.2.39, a–b)

Sei $\mathcal{E} = (\{S, B\}, \{b\}, S, R)$ mit $R = \{ S ::= \widehat{B}b, B ::= Sb \}$ eine EBNF-Definition. Berechnen Sie die syntaktischen Kategorien $W(\mathcal{E}, S)$ und $W(\mathcal{E}, B)$ mit Hilfe der Fixpunktsemantik. Gehen Sie dazu in den folgenden Schritten vor:

- Dokumentieren Sie mindestens 5 Iterationsschritte,
- schreiben Sie in Mengenschreibweise die Sprachen $W(\mathcal{E}, S)$ und $W(\mathcal{E}, B)$ auf, und
- zeigen Sie, dass $W(\mathcal{E}, S) = \llbracket \widehat{B}b \rrbracket(\rho)$ sowie $W(\mathcal{E}, B) = \llbracket Sb \rrbracket(\rho)$ erfüllt sind, wenn $\rho(S) = W(\mathcal{E}, S)$ und $\rho(B) = W(\mathcal{E}, B)$.

Zusatzaufgabe 2 (AGS 3.1.17)

Schreiben Sie ein Programm, das, für eine Eingabe n , den Nutzer eine zufällig bestimmte Zahl von 1 bis n raten lässt. Es soll nur ausgegeben werden, ob die geratene Zahl größer, kleiner, oder gleich der gesuchten Zahl ist.

Hinweis: Pseudo-zufällige Zahlen lassen sich mit der Funktion `rand()` generieren, dazu muss die Bibliothek `stdlib.h` eingebunden werden. Das folgende Programm gibt z.B. zehn pseudo-zufällige Zahlen aus.

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 int main () {
4     for (int i = 0; i < 10; i++)
5         printf("%d\n", rand());
6 }

```

Zusatz²aufgabe: Bewerten Sie den Nutzer im Vergleich zur optimalen Ratestrategie!