

3. Programmieren in C

Abgabe bis 21. Mai, 10:15 GMT+2.

Aufgabe 0:

Ändere dein Programm von Aufgabe 3, Blatt 1 so, dass es dynamische Speicherallokation benutzt (um die Größe von `marked` je nach Eingabe zu wählen)!

Aufgabe 1:

Schreibe eine Funktion `findX` in `find.c`, die das erste direkt auf ein 'a' folgende 'x' in einem als einzigen Argument übergebenen String findet, und einen Zeiger darauf zurückgibt. Falls kein 'x' gefunden wurde, soll 0 zurückgegeben werden. `findX` soll in `findTest.c` getestet werden.

Aufgabe 2:

Ändere deine Lösung von Aufgabe 2, Blatt 2 wie folgt! Deklariere in `cave.h` eine Struktur `cave`, in der sich die Adjanzenzmatrix, und das Array für die Fledermäuse befinden! Ersetze die Funktion `void init_cave(bool matrix[CAVESIZE][CAVESIZE], bool bats[CAVESIZE])` durch eine Funktion `void init_cave(struct cave *c)` in `cave.c`! `init_cave` soll in `cavetest.c` getestet werden.

Aufgabe 3:

Ergänze das Spiel `wumpus` aus Aufgabe 3, Blatt 2 wie folgt! Es wird `init_cave` aus Aufgabe 2 von diesem Blatt benutzt. Alle Funktionen (abgesehen von den Tests) sollen sinnvoll auf `cave.c`, `wumpus.c` und `wumpusMain.c` aufgeteilt sein. Funktionen in `cave.c` werden in `cavetest.c` getestet. Funktionen in `wumpus.c` werden in `wumpustest.c` getestet. In `wumpusMain.c` soll nur die Funktion `main` des Spiels stehen. Wenn der Spieler einen Raum mit Fledermäusen betritt, so packen diese den Spieler, tragen ihn in einen pseudozufälligen Raum, werden ihn dort ab (möglicherweise direkt dem Wumpus ins Maul) und kehren dann an ihren Ruheplatz zurück. Das Spiel `wumpus` soll beim Aufruf einen (optionalen) Parameter haben. Falls dieser angegeben wird, soll er eine Zahl sein, die mit `srand` den Pseudozufallszahlengenerator initialisiert.

Hinweise (gelten für dieses wie auch für alle weiteren Blätter):

- Die Abgabe erfolgt über Subversion in ein Unterverzeichnis Blatt3 (entsprechend bei den folgenden Blättern dann Blatt4, etc).
- Als Textkodierung für sämtliche Dateien ist UTF-8 zu verwenden. Die Verwendung von Abarten wie CESU-8, WTF-8 oder OMG-WTF-8 ist zu unterlassen.
- Soweit auf Aufgaben von vorigen Blättern Bezug genommen wird („Ändere Deine Lösung...“) sind korrekte Versionen der alten Lösung zu verwenden. Wenn die eigene Lösung also unzureichend war, so ist diese vor Verwendung zu verbessern, oder durch eine Musterlösung zu ersetzen. In letzterem Falle ist der Urheberrechtsvermerk aus der Musterlösung zu belassen, und um den eigenen zu ergänzen; das sähe dann z.B. so aus:

```
// Copyright 2019 Tutku Tutor  
// Copyright 2019 Sturla Student
```