



Systemnahe Software I (WS 2012/2013)

Abgabe bis zum 8. November 2012, 16:00 Uhr

Lernziele:

- Grundlegende Kontrollstrukturen (*if, for, while, ...*)
- Ein-/Ausgabe auf der Kommandozeile
- Erster Blick auf Arrays

Aufgabe 3: Nim-Spiel mit 3 Haufen

Ein Nim-Spiel ist ein Spiel für zwei Personen, bei dem jede abwechselnd eine Anzahl von Gegenständen, etwa Streichhölzer, von einer vorgegebenen Menge wegnehmen kann. Derjenige, der das letzte Hölzchen nehmen kann, gewinnt. Dieses Spiel hat zahlreiche Varianten und ist Gegenstand der kombinatorischen Spieltheorie.

Ziel dieser Aufgabe ist es nun, dieses Spiel in einer Variante mit 3 Haufen von Hölzern zu programmieren, sodass ein Spieler gegen den Computer spielen kann. Ein Zug besteht darin, von genau einem Haufen 1-3 Hölzchen zu entfernen. Bei der Umsetzung der Aufgabe sollen folgende Punkte berücksichtigt werden:

- Zu Spielbeginn soll die Anzahl der Hölzchen in jedem der 3 Haufen zufällig zwischen 5 und 10 bestimmt werden. Dazu wird die Funktion *rand* benötigt, die Zufallszahlen liefert. Mit *srand* kann ein Seed (=Startwert für den Pseudozufallszahlengenerator) gesetzt werden. Als Seeds bieten sich die momentane Zeit (*time*) oder die Prozess-ID (*getpid*) an.
- Der Spieler darf entscheiden, ob er als erster an die Reihe kommen will oder ob er dem Computer den Vorzug lassen will.
- Bei jedem Zug des menschlichen Spielers muss dieser den Haufen (1-3), von dem er ziehen will, und die Anzahl der Hölzchen (1-3), die er ziehen will, angeben.

- Falls eine Eingabe nicht erfolgreich war, d. h. nicht von *scanf* in eine *int*-Zahl umgewandelt werden konnte, soll das Programm mit einem Fehlercode terminiert werden (z. B. mit dem Systemaufruf *exit*).
- Bei Eingabe von ungültigen Zahlenwerten soll die Eingabe wiederholt werden können.
- Der Computer muss lediglich gültige Spielzüge durchführen, wenn er an der Reihe ist. Die Strategie ist dabei Euch überlassen und fließt nicht in die Bewertung mit ein.

Hinweis:

Die Verwendung von Arrays ist sehr nützlich zur Bearbeitung der Aufgabe, aber nicht zwingend erforderlich.

In C können Arrays, deren Länge zur Compile-Zeit feststeht, einfach durch Angabe der gewünschten Länge in eckigen Klammern hinter dem Namen des Arrays angelegt werden. Wie in Java sind die Einträge eines Arrays von 0 bis Länge -1 indiziert. Auf den Wert eines Eintrags eines Arrays kann ebenso wie in Java mit Angabe des Index in eckigen Klammern zugegriffen werden.

Folgender Beispielcode erzeugt ein *int*-Array der Länge 3 mit Namen „arr“ und weist jedem Eintrag des Arrays den Wert 0 zu:

```
int arr[3];
for (int i=0; i<3; ++i) {
    arr[i] = 0;
}
```

Der Submit-Befehl zum Einreichen der Lösung lautet:

```
thales$ submit ssl 3 nim3.c team
```

Viel Erfolg!