

# Schiffe versenken

Du erhältst ein Programm `schiffe_versenken.cpp`, das das Brettspiel „Schiffe versenken“ implementiert. Bitte kompiliere und starte dieses zunächst wie folgt:

```
g++ -o schiffe_versenken schiffe_versenken.cpp
./schiffe_versenken mensch mensch
```

Schiffe versenken ist ein Spiel für zwei Spieler (Spieler A und B). Jeder Spieler hat ein 10 x 10 Quadrate großes Spielfeld, durchnummeriert von Quadrat (0,0) links oben bis Quadrat (9,9) rechts unten. Zunächst wird **Spieler A** gefragt, 4 Schiffe auf sein Feld zu platzieren. Jedes Schiff beginnt bei einem Quadrat (zeile spalte) und belegt dann 4 Quadrate, entweder nach rechts (R) oder nach unten (U). Ein Beispiel für die Eingaben:

```
0 4 U
5 0 U
6 4 R
9 6 R
```

A	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

Das zugehörige Spielfeld findet sich rechts. Beachte: Die Schiffe dürfen sich weder horizontal noch vertikal berühren.

Danach ist **Spieler B** an der Reihe, der auch ein eigenes 10 x 10 Quadrate großes Spielfeld hat. Nun wird Spieler B gefragt (zur Zeit noch auf dem selben Terminal, was natürlich nicht ideal ist), wo er/sie die Schiffe platzieren möchte. Nehmen wir einmal beispielhaft folgende Eingaben an:

```
5 1 R
1 9 U
9 6 R
6 5 R
```

B	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

Das zugehörige Spielfeld findet sich wieder rechts.

Danach wechseln sich Spieler A und Spieler B ab. Zunächst in A an der Reihe: er gibt ein Quadrat in Spieler Bs Spielfeld ein, zum Beispiel:

```
5 3
```

Spieler A „schießt“ damit auf das jeweilige Quadrat von Spieler B.

Hat Spieler B in diesem Quadrat ein Schiff, so ist es ein Treffer, ansonsten ein Fehlschuss. Je nachdem gibt das Spiel T (für Treffer) oder F (für Fehlschuss) aus. Falls es ein Treffer ist das letzte Feld eines Schiffes getroffen wurde, gibt das Spiel statt T das Zeichen V (für Treffer und Versenkt) aus. In unserem Beispiel ist es ein Treffer, da B an dieser Stelle ein Schiff hat, und das Programm würde T ausgeben. Nun ist Spieler B an der Reihe usw.

Der erste Spieler, der alle 16 Quadrate, wo Schiffe sind, trifft, hat das Spiel gewonnen. In diesem Fall bekommt der Spieler, der gewonnen hat, im nächsten Zug anstatt T oder F als Ausgabe W (Gewinn) oder L (Verlust).

## Aufgabe

Schreibe ein Programm, das Schiffe versenken spielen kann, d.h. das die Aktionen eines Spielers (A oder B) in `schiffe_versenken` übernimmt. Das Programm kommuniziert über die Standardeingabe (`cin`) bzw. Standardausgabe (`cout`). Für alle anderen Ausgaben (Fehlersuche) verwende bitte `cerr`, oder noch besser, eine eigene Datei.

1. Zunächst schreibt das Programm auf `cout`, wo es die Schiffe platzieren möchte (siehe oben)
2. Danach soll es Züge ausführen:
  1. Das Programm schreibt zunächst auf `cout` das Quadrat, wo es hinschießen möchte, gefolgt von einem Zeilenumbruch.
  2. Das Programm liest dann von `cin` ein Zeichen ein. Dieses ist entweder T (Treffer), V (Treffer und Versenkt), F (Fehlschuss), W (gewonnen), L (verloren).
  3. Falls das Zeichen T, V oder F ist, verarbeitet es das Resultat und der nächste Zug wird ausgeführt. Fall das Zeichen W oder L ist, ist das Spiel vorbei und das Programm bricht ab.

Das Programm soll dabei halbwegs intelligent agieren, das heißt:

1. Es soll die Schiffe nicht immer an der selben Stelle platzieren, sondern die Positionen zufällig variieren. Hinweis: `#include<random>` ...
2. Es soll nach spätestens 70 Zügen alle Schiffe des Gegners versenkt haben.

Du kannst gegen Dein Programm spielen, indem Du anstatt `mensch` die ausführbare Datei Deines Programms angibst, d.h. falls Dein Programm zB `meine_ki` heißt:

```
./schiffe_versenken mensch ./meine_ki
```

Du kannst das Programm auch gegen sich selbst spielen lassen:

```
./schiffe_versenken ./meine_ki ./meine_ki
```