

### Crara springt

#### Musterlösung

a) kürzester Weg:

1	→ 2	1
2	1	2
2	↙ 1	→ 1

zweiter Weg:

1	→ 2	1
2	1	2
2	↖ 1	↓ 1

b) Es sind folgende drei Probleme zu erwarten:

- Wenn im mittleren Feld eine 2 steht, kann man von dort nicht weiter springen (kein Bild).
- Im oberen Bild kann (nur bei fehlerhafter Programmierung) eine Endlosschleife entstehen (durch den Pfeil angedeutet), obwohl eine Lösung existiert.
- Es kann sein, dass das Zielfeld gar nicht erreichbar ist, ein Beispiel ist im unteren Bild dargestellt.

1	→ 2	↑ 1
1	1	2
2	2	↓ 1

1	2	1
2	1	2
1	2	1

c) Die optimale Vorgehensweise besteht in der Nutzung des Backtracking-Verfahrens:

- Untersuchen, ob das Ziel erreicht ist.
- Wenn nicht, dann prüfen, ob das Trampolin schon betreten wurde.
- In alle vier Richtungen springen, wenn die Sprungweite nicht in den Abgrund führt.

d) Wenn die Sprungweite gleich der Kantenlänge (oder größer) ist, springt man immer in den Abgrund des Grauens, also bei einem 6x6-Feld darf man höchstens eine Sprungweite von 5 benutzen.

e) Lösung in der Datei 09R32.pas (Free Pascal)

f) Lösung in der Datei 09R32.pas (Free Pascal)

#### Musterlösung

09R32.pas

#### Punktverteilung

- |    |   |   |
|----|---|---|
| a) | Darstellung oder Beschreibung des kürzesten Weges und eines zweiten Weges | 2 |
| b) | 2 Probleme und 2 Begründungen   | 4 |
| c) | sinnvoller Algorithmus  | 2 |
|    | Rekursionsabbruch bei Erreichen des Zieles                                | 1 |
|    | Rekursionsabbruch bei Endlosschleifen bzw. Unlösbarkeit                   | 1 |
| d) | Sprungstärke 5; Begründung  | 2 |
| e) | Umsetzung des Algorithmus aus Aufgabe c)                                  | 3 |
|    | Berücksichtigung der Sonderfälle  | 3 |
| f) | Computerlösung, die den kürzesten Weg findet                              | 2 |

**Gesamt**

**20**