

Aufgabe 1: Superstar

Symbroson
Team-ID: 00165

9. September 2018

Inhaltsverzeichnis

1	Lösungsidee	1
2	Umsetzung	1
3	Quellcode	1

1 Lösungsidee

Diese Aufgabe kann am besten durch Brute-Force gelöst werden. Dh. es wird jede Würfelzahl durchprobiert, ob der Spieler bei dieser das Ziel erreicht oder nicht. Um zu erkennen, ob man das Ziel nicht erreichen kann, muss geprüft werden, ob man das gerade betretene Feld bereits betreten hat. Falls ja, muss der Prozess abgebrochen werden, da man sonst in eine Schleife gerät.

2 Umsetzung

Zunächst müssen alle Leitern katalogisiert werden. Da es keine Eingabedatei gibt sondern nur ein einziges Spielfeld kann dies durch ein konstantes Array geschehen, in dem jeweils Anfang und Ende jeder Leiter fortlaufend gespeichert werden.

In einem zweiten Array werden aller Spielfelder durch eine Klasse dargestellt, die den 'bereits betreten' Status und ggf. das Zielfeld der Leiter enthält.

Nun muss für jede Würfelzahl solange die 'Spielfigur' gesetzt werden, bis diese am Ziel angelangt oder auf ein bereits betretenes Feld gelangt. Dies wird dann dementsprechend mit der Anzahl der Züge tabellarisch ausgegeben.

3 Quellcode

```
#define LEDDER_COUNT 12
#define FIELD_COUNT 100
#define FIELD(p) fields[p - 1]

5 struct Field {
    bool touched = false; // Status: Feld betreten
    uint8_t target = 0;    // ggf. Ziel der Leiter
} fields[FIELD_COUNT + 1];

10 // Leitern
uint8_t ledders[LEDDER_COUNT * 2] = {6, 27, 14, 19, 21, 53, 31, 42,
                                     33, 38, 46, 62, 51, 59, 57, 96,
                                     65, 85, 68, 80, 70, 76, 92, 98};

15 int main(int argc, const char* argv[]) {
    uint8_t n, // Würfelzahl
    p,       // Position
    i;       // Zähler
```

Aufgabe 1: Superstar

```
20     uint32_t c; // Wurfanzahl

    // Wende Leiterliste an
    for (i = 0; i < LEDDER_COUNT * 2; i += 2) {
        FIELD(ledders[i]).target = ledders[i + 1];
25     FIELD(ledders[i + 1]).target = ledders[i];
    }

    printf(
        "Zahl | Ziel | Würfe\n"
30     "-----+-----\n");

    // alle Würfelzahlen durchprobieren
    for (n = 1; n <= 6; n++) {
        p = 1;
35     c = 0;

        // resette Felder
        for (i = 0; i < FIELD_COUNT; i++) fields[i].touched = false;

40     // solange Ziel nicht erreicht
        while (p != FIELD_COUNT) {
            p += n;
            c++;

45     // 100 überschritten
            if (p > FIELD_COUNT - 1) p = 2 * FIELD_COUNT - p;

            // ggf. Leiter benutzen
            if (FIELD(p).target) p = FIELD(p).target;

50     // Feld bereits betreten? -> abbrechen
            if (FIELD(p).touched) break;

            FIELD(p).touched = true;

55     }

        printf("%4i | %4s | %4i\n", n, p == FIELD_COUNT ? "ja" : "nein", c);
    }
}
```