

## Kassiopeia

In Quadratien ist das Land in schwarze und weiße quadratische Felder einheitlicher Größe aufgeteilt. Auf jedem weißen Feld steht eine Pflanze, aus der täglich ein Blatt sprießt. Die Schildkröte Kassiopeia lebt schon seit vielen Jahren von diesen Blättern. Jeden Tag muss sie alle Blätter fressen, um zu überleben. Also muss sie jedes weiße Feld besuchen. Die schwarzen Felder heizen sich unter der Sonne so sehr auf, dass sie diese nicht betreten kann. Von einem weißen Feld kann sie in einem Schritt zu jedem in nördlicher, südlicher, östlicher oder westlicher Richtung benachbarten weißen Feld gelangen.

Schreibe ein Programm, das für eine Version von Quadratien ausgibt, ob Kassiopeia alle weißen Felder erreichen kann, ohne ein schwarzes Feld zu überqueren.

Eine Startsituation kann folgendermaßen aussehen:

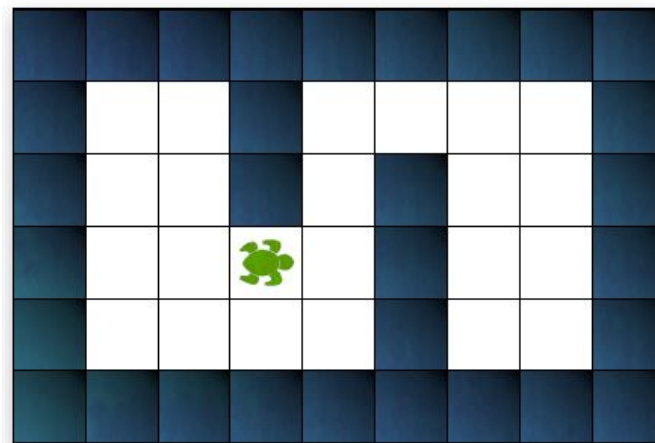


Figure 1: Beispielfeld Kassiopeia

Die String-Repräsentation für dieses Feld wäre:

```
field = [
    ['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X'],
    ['X', '_', '_', 'X', '_', '_', '_', '_', 'X'],
    ['X', '_', '_', 'X', '_', '_', '_', '_', 'X'],
    ['X', '_', '_', '_', '_', 'X', '_', '_', 'X'],
    ['X', '_', '_', '_', '_', 'X', '_', '_', 'X'],
    ['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X']
]
```

Jeder Eintrag der Matrix `field` ist hierbei entweder ein `_` für ein weißes Feld ist oder ein `X` für ein schwarzes Feld.

Implementieren Sie folgende Funktionen:

- `def get_neighbors(x: int, y: int, field_max_x: int, field_max_y: int) -> List[Tuple[int, int]]:`

Diese Funktion soll die Koordinaten aller Nachbarmfelder des Feldes `(x, y)` berechnen. In den beiden Argumenten `field_max_x` und `field_max_y` ist die maximale Breite und Höhe des Feldes gegeben.

- `def test_survivable(start: Tuple[int, int], field: List[List[str]]) -> bool:`

Diese Funktion soll berechnen, ob die Schildkröte auf dem übergebenen Feld `field` überleben kann. Dieses ist im oben beschriebenen Format. Das Argument `start` ist die Startkoordinate der Schildkröte. Gehen Sie dabei folgendermaßen vor (in Anlehnung an die Breitensuche):

- Es wird eine Liste von Feldern erstellt, die zuerst nur das Startfeld beinhaltet
- Solange diese Liste nicht leer ist
  - \* Wird das erste Element entnommen
  - \* Alle Nachbarn dieses Elements berechnet
  - \* Jeder Nachbar, der noch nicht besucht wurde (`_`), wird als besucht markiert (z. B. `V` in der Matrix) und in die Liste eingefügt
- Sobald die Liste leer ist, wird geprüft, ob es noch nicht besuchte Felder gibt. Diese sind nicht erreichbar. Damit kann die Schildkröte nicht überleben.