

Der Vorkurs ist ein Angebot der (teilweise ehemaligen) Programmierung 2 Tutoren. Der Kurs ist keine offizielle Lehrveranstaltung. Es gibt keine CP, die Teilnahme ist optional, und eine HISPOS-Anmeldung ist weder möglich noch erforderlich. Bei Fragen zum Vorkurs und Programmierung 2 könnt ihr euch gerne an die Dozenten und Tutoren wenden. Wir wünschen euch ein erfolgreiches Semester und freuen uns, euch in Programmierung 2 wieder zu sehen.

Aufgabe 2.0: Listen

In diesem Task geht es um Listen (arrays). In dem vorhandenen Code wird bereits eine Abfolge von Richtungen in das Array geschrieben. Diese Abfolge von Bewegungen wird unsere Eule zum Ziel bringen. Ihre Aufgabe ist es nun, ein Programm zu schreiben, dass die Richtungen einzeln extrahiert und einen Schritt in diese Richtung macht. Zur Hilfe ist die Länge des Arrays in die Variable `length` geschrieben worden.

```
arr_set 0 left
arr_set 1 left
arr_set 2 front
arr_set 3 left
arr_set 4 front
arr_set 5 front
arr_set 6 front
arr_set 7 left
arr_set 8 front
arr_set 9 front
arr_set 10 right
arr_set 11 left
arr_set 12 front

var int length
set length 13
```

Aufgabe 2.1: Rechte-Hand-Regel

In der Vorlesung haben Sie bereits die Linke-Hand-Regel kennen gelernt. Ihre Aufgabe ist nun die Rechte-Hand-Regel zu implementieren, das heißt: Laufen Sie nach rechts wenn immer möglich, ansonsten geradeaus. Wenn das Feld geradeaus auch nicht frei ist, drehen Sie sich nach links und versuchen Sie es erneut. Gegeben ist der Code für die Linke-Hand-Regel aus der Vorlesung.

```
var int noWallLeft
var int noWallFront

next_move:
  explore noWallLeft left
  branch noWallLeft no_wall_to_left
  jump wall_to_left

no_wall_to_left:
  turn left
  move
  jump next_move

wall_to_left:
  explore noWallFront front
```

```

branch noWallFront no_wall_in_front
jump wall_in_front

no_wall_in_front:
move
jump next_move

wall_in_front:
turn right
jump next_move

```

Aufgabe 2.2: Pledge

Einfach nur der Wand zu folgen, funktioniert leider nicht immer. Deswegen lernen wir jetzt noch eine Erweiterung der rechten Hand Regel: den Pledge Algorithmus.

Solange du nicht am Ziel bist:

Setze den Umdrehungszähler auf 0

Solange kein Hindernis vor dem Spieler liegt:

Gehe gerade aus

Ist der Spieler auf ein Hindernis gestoßen:

drehe dich nach rechts

erhöhe den Umdrehungszähler um 1

Solange der Umdrehungszähler $\neq 0$

Folge dem Hindernis mit der linken Hand Regel

Erhöhe den Umdrehungszähler für jede Rechtsdrehung um 1

Erniedrige ihn für jede Linksdrehung um 1

Dann gehe wieder gerade aus bis zum nächsten Hindernis

Aufgabe 2.3: Bonus: Dead-End-Filling

Unsere Eule hat sich wieder im Labyrinth verirrt und braucht deine Hilfe um raus zu kommen. Allerdings will sie nicht wieder Wänden folgen, weil sie das langweilig findet und stattdessen folgenden Algorithmus ausprobieren:

Solange du nicht am Ziel bist:

Wenn du aus einer Sackgasse kommst:

markiere den Weg hinter dir

Überprüfe nun, ob du in einer Sackgasse bist

Wenn ja:

gehe den einen Weg der noch möglich ist

Sonst:

gehe einen beliebigen möglichen Weg

(Eine Sackgasse ist ein Feld, das ≤ 1 passierbaren Nachbar hat) Implementieren Sie nun diesen Algorithmus, um die Eule zum Ziel zu führen.