

The Task: Check a Maze for Cycles

The task: check for cycles in a maze

- using our earlier maze representation:

```
static uint8_t maze[10][10]; // maze
// L = 1, R = 2, U = 4, D = 8
```

- Is a cycle reachable from starting point?
- You define the function signature.
- You probably should use (initialized to 0):

```
static uint8_t found[10][10];
```

Review: How the Bit Vector Representation Works

We can **represent the maze with an array**:

```
static uint8_t maze[10][10];
```

Each space in the array **is a bit vector** composed of the following bits:

- // **1** – the space has a **left wall**
- // **2** – the space has a **right wall**
- // **4** – the space has an **upper wall**
- // **8** – the space has a **lower wall**
- // **16** – the space is the **exit**

Tasks on Your Own

More things to try on your own:

1. Remove all cycles (by adding walls as necessary).
2. Make all parts of the maze connected (by removing walls as necessary).