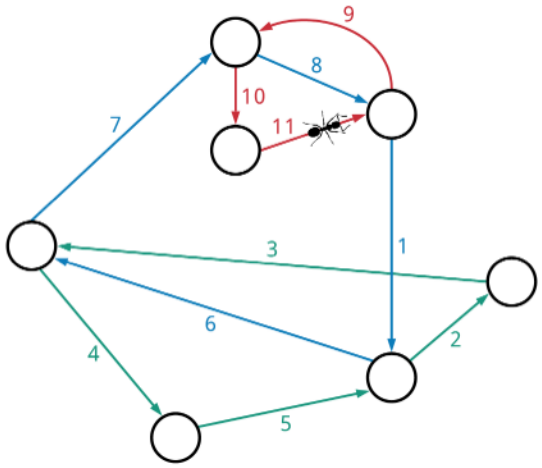
## 3F Find an Eulerian Cycle in a Graph

**Eulerian Cycle Problem**
*Find an Eulerian cycle in a graph.*

**Input:** An Eulerian directed graph.
**Output:** An Eulerian cycle in this graph.




### Formatting

**Input:** An adjacency list representing an Eulerian directed graph.
**Output:** A space-separated list of integers representing an Eulerian cycle in the directed graph.

### Constraints

- The number of nodes in the graph will be between 1 and $10^4$.
- The number of edges in the graph will be between 1 and $10^4$.
- All nodes in the graph will be labeled with integers.

## Test Cases

### Case 1

**Description:** The sample dataset is not actually run on your code.

**Input:**

```
0:  3
1:  0
2:  1 6
3:  2
4:  2
5:  4
6:  5 8
7:  9
8:  7
9:  6
```

**Output:**

```
3 2 6 8 7 9 6 5 4 2 1 0 3
```

### Case 2

**Description:** The sample dataset is not actually run on your code.

**Input:**

```
0:  1
1:  2
2:  0
```

**Output:**

```
0 1 2 0
```

### Case 3

**Description:** The sample dataset is not actually run on your code.

**Input:**

```
0:  3 1
1:  2
2:  0
3:  0
```

**Output:**

```
0 3 0 1 2 0
```

**Case 4**

**Description:** The sample dataset is not actually run on your code.

**Input:**

```
0:  1
1:  2 3
2:  0
3:  4
4:  1
```

**Output:**

```
4 1 2 0 1 3 4
```

**Case 5**

**Description:** The sample dataset is not actually run on your code.

**Input:**

```
1:  2
2:  1 2
```

**Output:**

```
2 2 1 2
```

**Case 6**

**Description:** The sample dataset is not actually run on your code.

**Input:**

```
1:  10
10:  2 3 4
2:  1
3:  10
4:  5
5:  10
```

**Output:**

```
1 10 4 5 10 3 10 2 1
```

**Case 7**

**Description:** The sample dataset is not actually run on your code.

**Input:**

```
0:  1 2 3 4
1:  0 2 3 4
2:  0 1 3 4
3:  0 1 2 4
4:  0 1 2 3
```

**Output:**

```
3 4 3 1 3 0 2 0 4 0 3 2 1 0 1 2 4 1 4 2 3
```

**Case 8**

**Description:** A larger dataset of the same size as that provided by the randomized autograder. Check input/output folders for this dataset.