

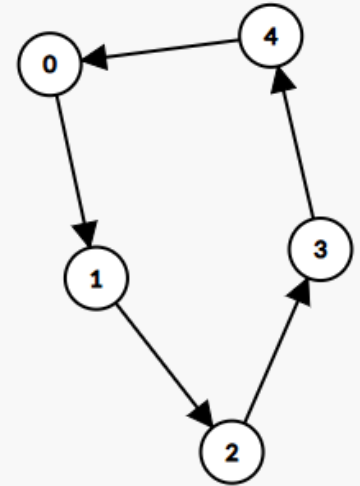
Name :	ID:	Section:
--------	-----	----------

### Question [15 Points]

Write a Python or Java function called **wantToReturnStart**. The function takes **the given graph** represented as an adjacency matrix (a 2D list where  $\text{graph}[i][j] = 1$  means there is a directed edge from node  $i$  to node  $j$ , and  $\text{graph}[i][j] = 0$  means there is no edge from node  $i$  to node  $j$ ).

For the given graph, the function should return **True** if there exists any cycle (a cycle is defined as a path where you can start from **node 0**, follow some edges, and eventually return to **node 0**) and **False** otherwise.

**Note:** You have to manually build an adjacency matrix at first to represent the following directed unweighted graph. Also, you cannot use any built-in function, tuple, or dictionary. But you may build any helper function. And, assume that any code portion is not implemented.

Given Graph	Input of function	Output	Explanation
	graph [as an adjacency matrix]	True	The graph provided is a directed graph where node 0 can reach itself through the path $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0$ (using 2D list traversal)

*\*Hint: Solving this question is much easier than getting out of bed this winter morning.*