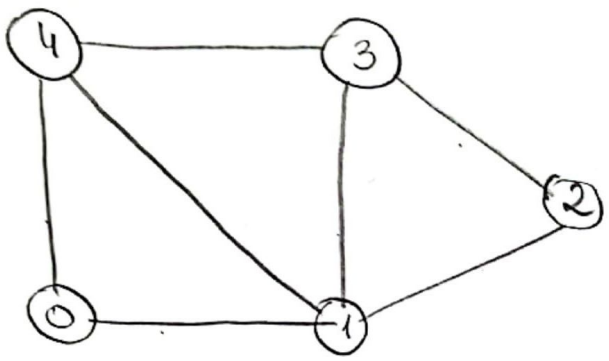

Input file :

5 7
0 1
0 4
1 2
1 3
1 4
2 3
3 4

Iteration	position	path	current_vertex	previous_vertex
init	—	[0, -1, -1, -1, -1]	—	—
i1.1	1	[0, 1, -1, -1, -1]	1	0 -1 => break; => True edge(0,1) in graph => is_safe = True
i2.1	2		1	0 1 => is_safe = False

Iteration	position	path	current_vertex	previous_vertex
i 2.2	2	[0, 1, -1, -1, -1]	2	0 1 -1 => break => True edge (2,1) in graph => is_safe = True
		[0, 1, 2, -1, -1]		
i 3.1	3	[0, 1, 2, -1, -1]	0	0 => is_safe = False
i 3.2			1	0 1 => is_safe = False
i 3.3			2	0 1 2 => is_safe = False
i 3.4			3	0 1 2 -1 => break => True

Iteration	position	path	current_vertex	previous_vertex
i 3.4	3	[0, 1, 2, -1, -1]	3	... edge (3, 2) in graph ⇒ is_safe = True
		[0, 1, 2, 3, -1]		
i 4.1	4	[0, 1, 2, 3, -1]	0	0 ⇒ is_safe=False
i 4.2			1	0
i 4.3				1 ⇒ is_safe=False
			2	0
				1
i 4.4				2 ⇒ is_safe=False
			3	0
				1
				2
				3 ⇒ is_safe=False

Iteration	position	path	current_vertex	previous_vertex	
i 4.5	4	[0, 1, 2, 3, -1]	4	0 1 2 3 -1 → Break → True edge (4, 3) in graph → is_safe = True	
		[0, 1, 2, 3, 4]			
i 5.1	5 position = no_vertices; edge (4, 0) in graph ⇒ True	[0, 1, 2, 3, 4]			

At this point all previous recursive calls of <__ find_hamiltonian_cycle_util> will return True.

The path [0, 1, 2, 3, 4] will be returned, to which we append the first vertex ⇒ [0, 1, 2, 3, 4, 0]