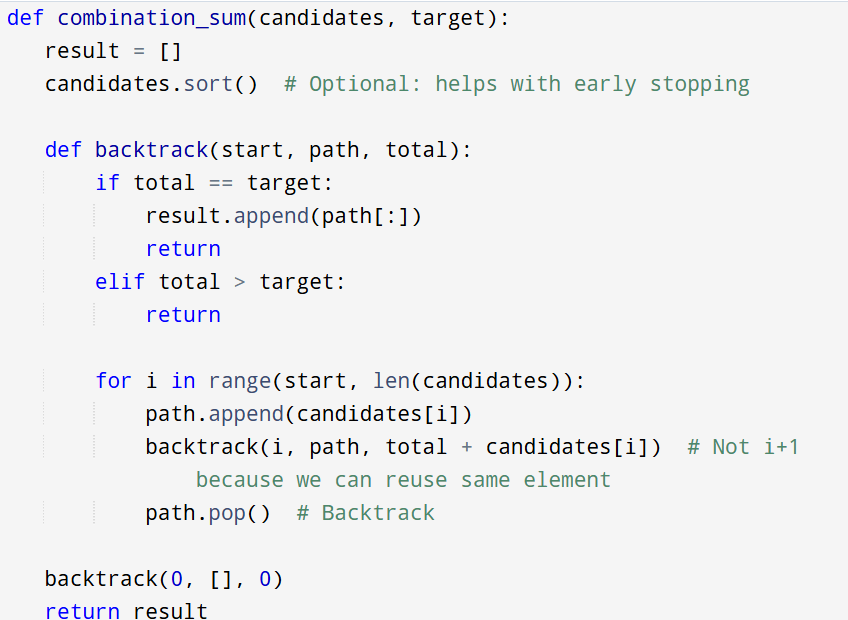
**7.1Hamiltonian Path**

**Aim:** To construct a python code to solve the Hamiltonian Path problem.

**Algorithm:**

1. Implement a program to verify if a given decision problem is in P or NP.
2. Selected Problem: Hamiltonian Path
3. Definition: A Hamiltonian Path is a path in a graph that visits each vertex exactly once.
4. This problem is in NP — there's no known polynomial-time algorithm to solve it for general graphs, but given a proposed path, we can verify it in polynomial time.

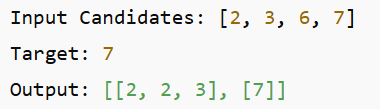
**Program:**



**Input :**  vertices = ['A', 'B', 'C', 'D']

edges = [('A', 'B'), ('B', 'C'), ('C', 'D'), ('D', 'A')]

**Output:**

****

**Result:** Program is been executed.

**Performance analysis:**

* Time complexity: O(2^t)
* Space complexity: O(t)