Assignment 3.1 Practice Problem 1 (Build a Graph)

Problem

You are given an integer n. Determine if there is an unconnected graph with n vertices that contains at least two connected components and contains the number of edges that is equal to the number of vertices. Each vertex must follow one of these conditions:

1. Its degree is less than or equal to 1
2. It's a cut-vertex

Note:

- The graph must be simple.
- Loops and multiple edges are not allowed.

Input format:

- First line: n

Output format:

- Print Yes if it is an unconnected graph. Otherwise, print No.

Constraints

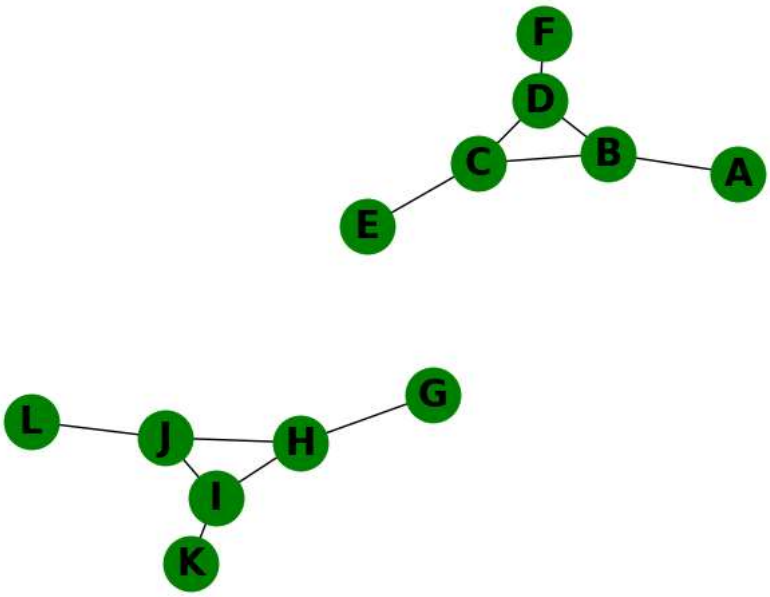
- 1 <= n <= 100

```
import networkx as nx


# Define the graph
Graph = {
    "A": {"B"},
    "B": {"A", "C", "D"},
    "C": {"B", "D", "E"},
    "D": {"B", "C", "F"},
    "E": {"C"},
    "F": {"D"},
    "G": {"H"},
    "H": {"G", "I", "J"},
    "I": {"H", "J", "K"},
    "J": {"H", "I", "L"},
    "K": {"I"},
    "L": {"J"}
}

# Create a directed graph object
G = nx.Graph(Graph)

# Draw the graph
nx.draw(G, with_labels=True, node_size=850, node_color='green', font_size=20, font_weight='bold')
```




```
n = int(input("Enter number of vertices: "))

def graph_connection(n):
  if n < 1:
    print("No")

  if 1 <= n and n <= 100:
    if n < 12:
      print("No")
    else:
      print("Yes")

  if n > 100:
    print("No")
graph_connection(n)
```

```
Enter number of vertices: 101
No
```