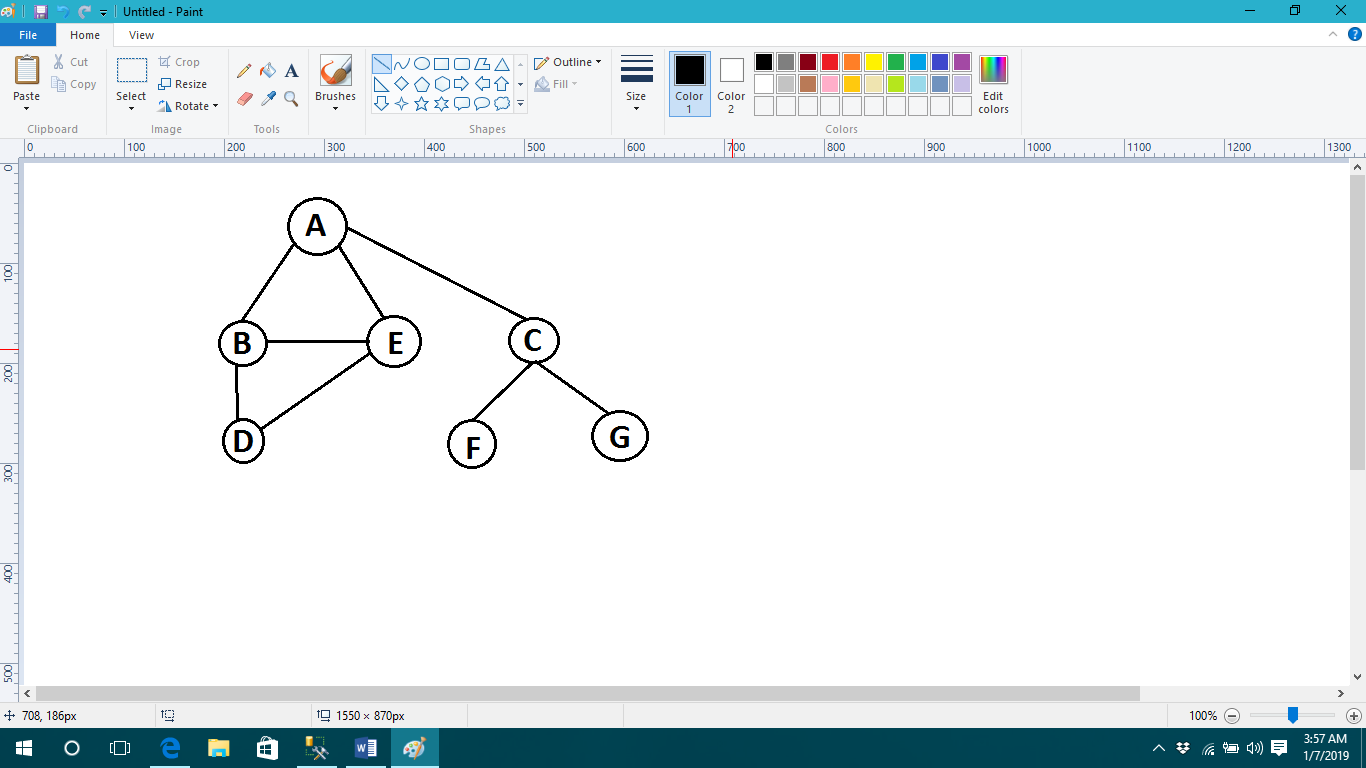
**IMPLEMENTATION OF BREATH FIRST SEARCH IN PYTHON**

**EXERCISE:**

**Object 01:**

Consider the following graph:



Represent a graph by using any data structure .Apply BFS if the starting node is “A” and goal node is “G” print complete path from source to destination.

**Source Code:**

graph = {'A': ['B', 'C', 'E'],

'B': ['A','D', 'E'],

'C': ['A', 'F', 'G'],

'D': ['B','E'],

'E': ['A', 'B','D'],

'F': ['C'],

'G': ['C']}

def bfs(graph, start):

fifo = []

queue = [start]

visited= [start]

while queue:

node = queue.pop(0)

fifo.append(node)

path = graph[node]

for i in path:

if i not in visited:

queue.append(i)

visited.append(i)

return fifo

path1 = bfs(graph,'A')

print(path1)

**Output:**

['A', 'B', 'C', 'E', 'D', 'F', 'G']

**Object 02:**

By using graph ,ask user to state starting and goal node .Apply BFS and print complete path from source to destination.

**Source Code:**

graph = {'A': ['B', 'C', 'E'],

'B': ['A','D', 'E'],

'C': ['A', 'F', 'G'],

'D': ['B','E'],

'E': ['A', 'B','D'],

'F': ['C'],

'G': ['C']}

def bfs(graph, start,final):

fifo = []

queue = [start]

visited= [start]

while queue:

node = queue.pop(0)

fifo.append(node)

path = graph[final]

for i in path:

if i not in visited:

queue.append(i)

visited.append(i)

return fifo

msg=input("Enter your initial point: ")

msg1=input("Enter your destination point: ")

path1 = bfs(graph,msg,msg1)

print(path1)

**Output:**

Enter your initial point: A

Enter your destination point: D

['A', 'B', 'E', 'D']