Practical no 1

AIM: Implement Breadth first search algorithm for Romanian map problem or any other map

Code:

```
from collections import defaultdict
class Graph:
    def __init__(self):
        self.graph = defaultdict(list)
    def addEdge(self,u,v):
        self.graph[u].append(v)
    def BFS(self, s):
        visited = [False] * (len(self.graph))
        queue = []
        queue.append(s)
        visited[s-1] = True
        while queue:
            s = queue.pop(0)
            print (s, end = " ")
            for i in self.graph[s]:
                if visited[i-1] == False:
                    queue.append(i)
                    visited[i-1] = True
g = Graph()

g.addEdge(1, 2)
g.addEdge(1, 3)
g.addEdge(2, 1)
g.addEdge(2, 4)
g.addEdge(2, 5)
g.addEdge(3, 1)
g.addEdge(3, 5)
g.addEdge(4, 2)
g.addEdge(4, 6)
g.addEdge(5, 2)
g.addEdge(6, 5)

print ("Following is Breadth First Traversal"

" (starting from vertex 1)")

g.BFS(1)

```

output:

```
========= RESTART: C:\Users\BlackBot\Desktop\tycs prac\al prac\pracl.py ========
Following is Breadth First Traversal (starting from vertex 1)
1 2 3 4 5 6
performed by krunal
>>>
```