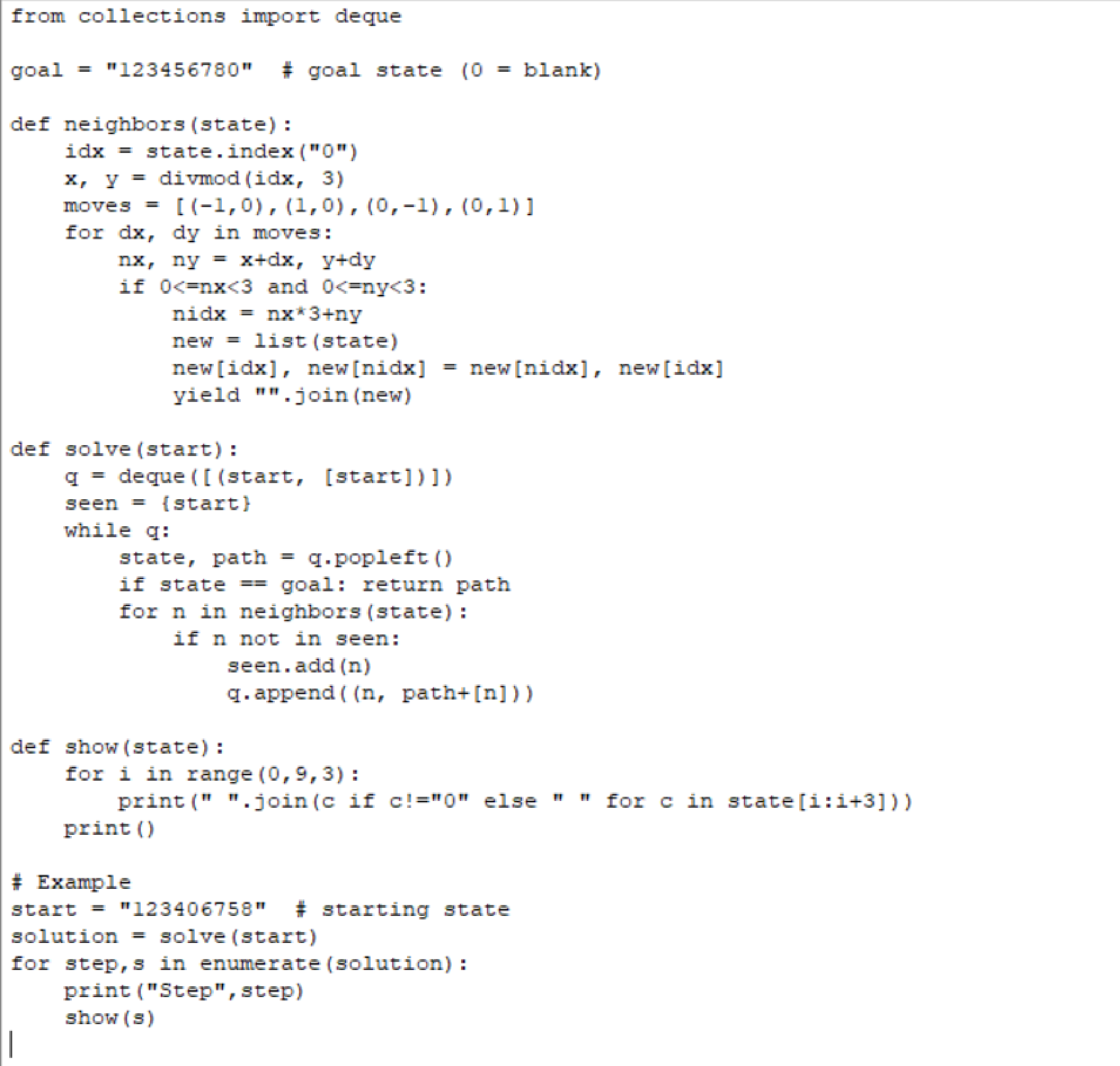
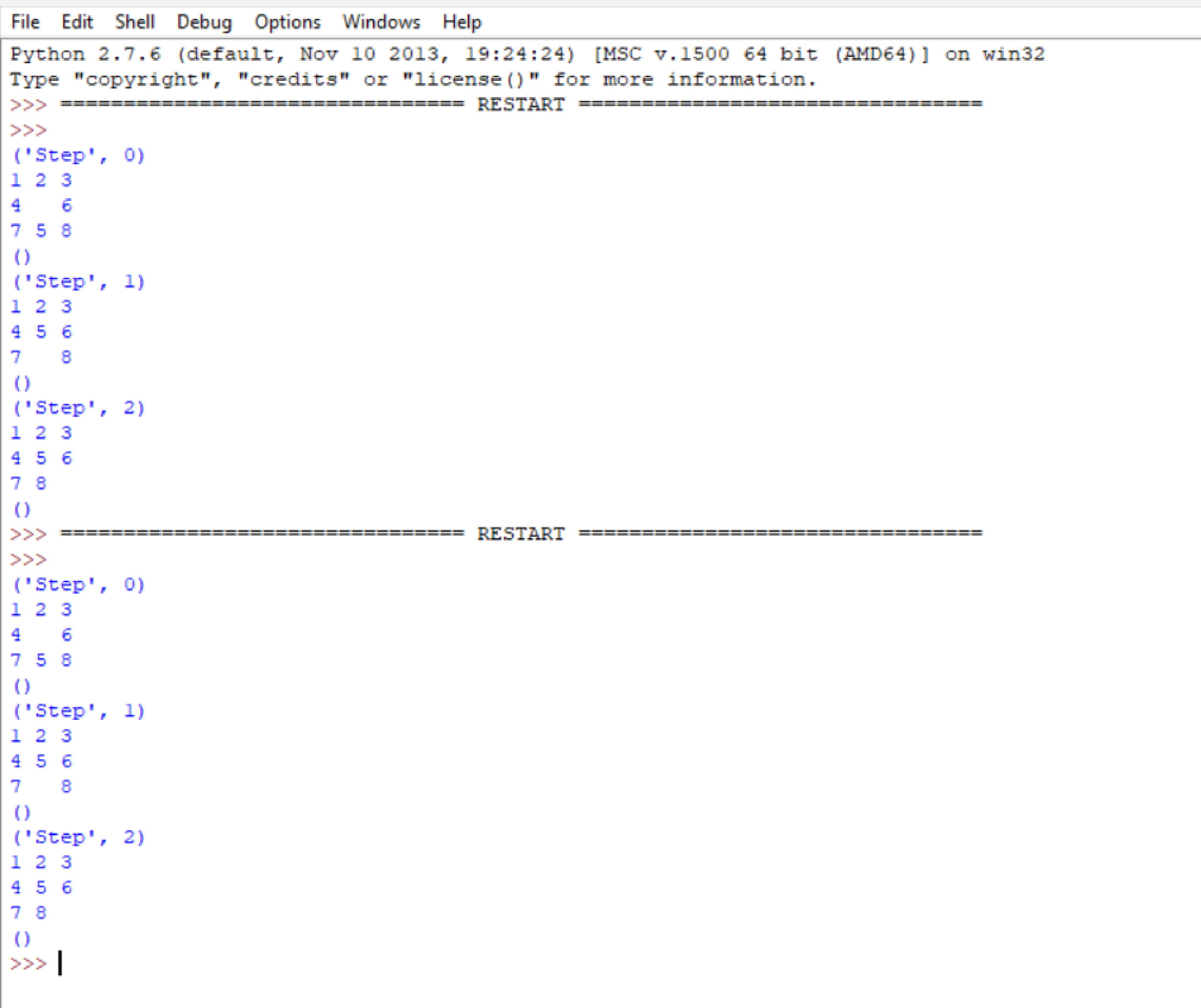


```python
from collections import deque

goal = "123456780"  # goal state (0 = blank)

def neighbors(state):
    idx = state.index("0")
    x, y = divmod(idx, 3)
    moves = [(-1,0),(1,0),(0,-1),(0,1)]
    for dx, dy in moves:
        nx, ny = x+dx, y+dy
        if 0<=nx<3 and 0<=ny<3:
            nidx = nx*3+ny
            new = list(state)
            new[idx], new[nidx] = new[nidx], new[idx]
            yield "".join(new)

def solve(start):
    q = deque([(start, [start])])
    seen = {start}
    while q:
        state, path = q.popleft()
        if state == goal: return path
        for n in neighbors(state):
            if n not in seen:
                seen.add(n)
                q.append((n, path+[n]))

def show(state):
    for i in range(0,9,3):
        print(" ".join(c if c!="0" else " " for c in state[i:i+3]))
    print()

# Example
start = "123406758"  # starting state
solution = solve(start)
for step,s in enumerate(solution):
    print("Step",step)
    show(s)
```



```
Python 2.7.6 (default, Nov 10 2013, 19:24:24) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
('Step', 0)
1 2 3
4   6
7 5 8
()
('Step', 1)
1 2 3
4 5 6
7   8
()
('Step', 2)
1 2 3
4 5 6
7 8  
()
>>> ================================ RESTART ================================
>>> 
('Step', 0)
1 2 3
4   6
7 5 8
()
('Step', 1)
1 2 3
4 5 6
7   8
()
('Step', 2)
1 2 3
4 5 6
7 8  
()
>>> 
```