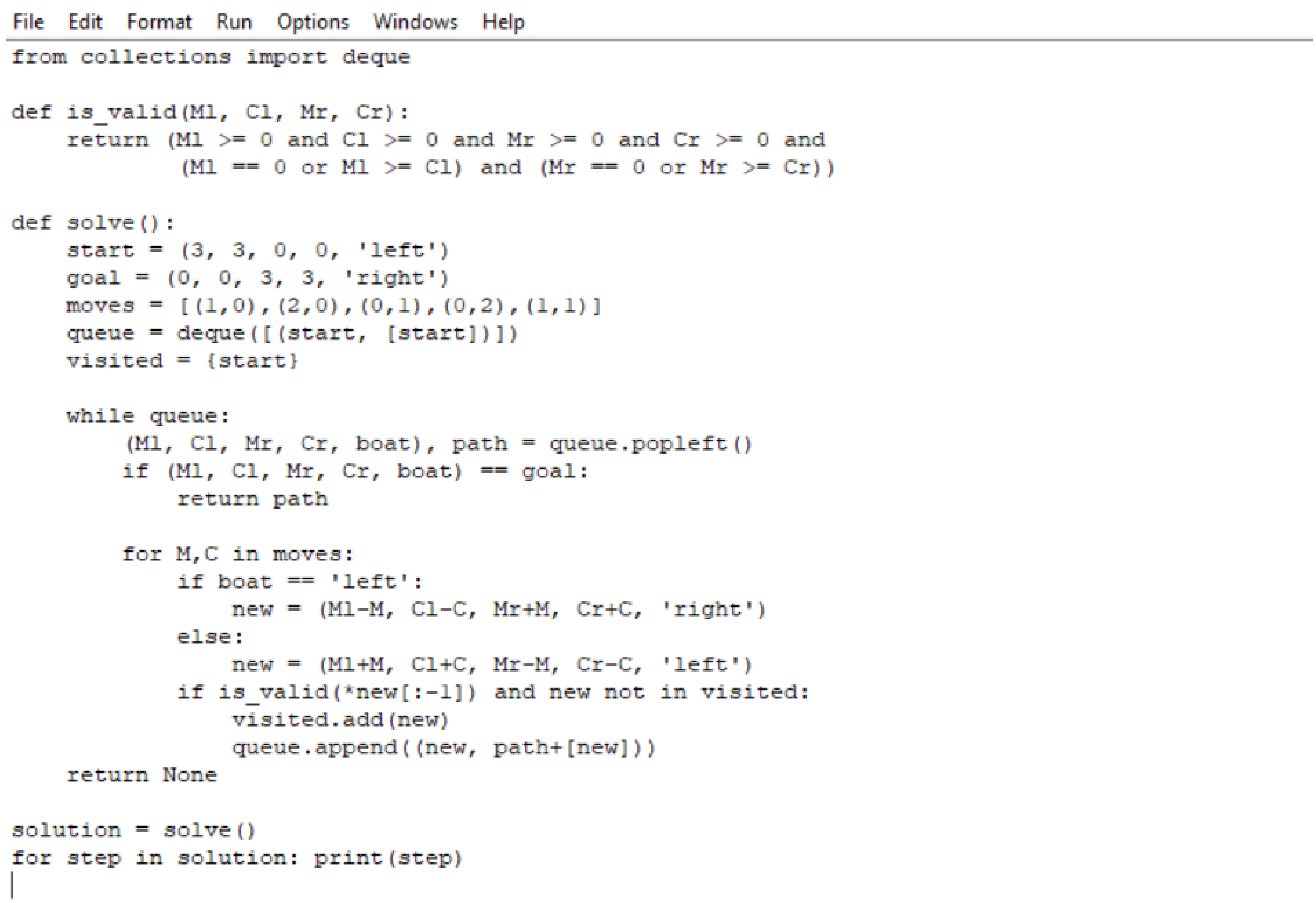
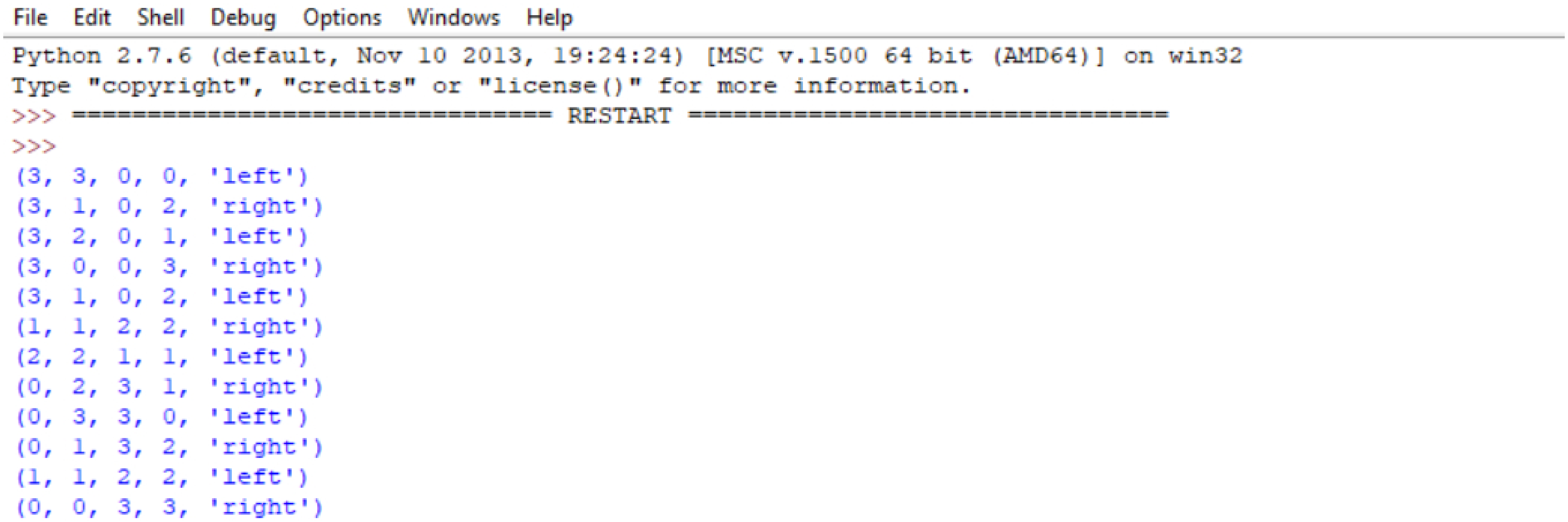


```python
from collections import deque

def is_valid(Ml, Cl, Mr, Cr):
    return (Ml >= 0 and Cl >= 0 and Mr >= 0 and Cr >= 0 and
            (Ml == 0 or Ml >= Cl) and (Mr == 0 or Mr >= Cr))

def solve():
    start = (3, 3, 0, 0, 'left')
    goal = (0, 0, 3, 3, 'right')
    moves = [(1,0),(2,0),(0,1),(0,2),(1,1)]
    queue = deque([(start, [start])])
    visited = {start}

    while queue:
        (Ml, Cl, Mr, Cr, boat), path = queue.popleft()
        if (Ml, Cl, Mr, Cr, boat) == goal:
            return path

        for M,C in moves:
            if boat == 'left':
                new = (Ml-M, Cl-C, Mr+M, Cr+C, 'right')
            else:
                new = (Ml+M, Cl+C, Mr-M, Cr-C, 'left')
            if is_valid(*new[:-1]) and new not in visited:
                visited.add(new)
                queue.append((new, path+[new]))
    return None

solution = solve()
for step in solution: print(step)
```



```
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 ================================
>>> 
(3, 3, 0, 0, 'left')
(3, 1, 0, 2, 'right')
(3, 2, 0, 1, 'left')
(3, 0, 0, 3, 'right')
(3, 1, 0, 2, 'left')
(1, 1, 2, 2, 'right')
(2, 2, 1, 1, 'left')
(0, 2, 3, 1, 'right')
(0, 3, 3, 0, 'left')
(0, 1, 3, 2, 'right')
(1, 1, 2, 2, 'left')
(0, 0, 3, 3, 'right')
```