```python
from puzzles import Puzzle8
from copy import deepcopy
from queue import PriorityQueue
'''
h1 = the number of misplaced tiles. For Figure 3.28,
all of the eight tiles are out of position, so the
start state would have h1 = 8. h1 is an admissible
heuristic because it is clear that any tile that is
out of place must be moved at least once.
'''
"""
puzzles =[]
for i in range(100):
    puzzles.append(Puzzle8.Puzzle())
"""


q = PriorityQueue()
explored = {""}
cost = 0
y = Puzzle8.Puzzle()
z = Puzzle8.Puzzle(shuffle=True)



x = y

if False:
    x.puzzle = [8, 6, 7, 2, 5, 4, 3, 0, 1]#[5, 1, 4, 6, 3, 8, 0, 7, 2]  # [3, 6, 2, 5, 0, 7, 
4, 1, 8]
    x.distCheck()
    x.findIndex()

explored.add(str(x.puzzle))

while x._dist != 0 and cost < 2000000:
    up = deepcopy(x)
    down = deepcopy(x)
    left = deepcopy(x)
    right = deepcopy(x)

    x1 = up.up()
    x2 = down.down()
    x3 = left.left()
    x4 = right.right()

    if x1 and str(up.puzzle) not in explored:
        q.put(up)
        explored.add(str(up.puzzle))
        up.parent_node = x
        #print(up, up.puzzle, "up", up._index, "index", "\n\n")
    if x2 and str(down.puzzle) not in explored:
        q.put(down)
        explored.add(str(down.puzzle))
        down.parent_node = x
        #print(down, down.puzzle, "down", down._index, "index", "\n\n\n")
    if x3 and str(left.puzzle) not in explored:
        q.put(left)
        explored.add(str(left.puzzle))
        left.parent_node = x
        #print(left, left.puzzle, "left", left._index, "index", "\n\n\n")
    if x4 and str(right.puzzle) not in explored:
        q.put(right)
        explored.add(str(right.puzzle))
        right.parent_node = x
        #print(right, right.puzzle, "right", right._index, "index", "\n\n")
    x = q.get()
    x._globalCost += 1

    if cost % 100 == 0:
        print(cost)
    #print(x._dist, " -------", x._globalCost)
    cost += 1


temp = x
lst = []
while temp.parent_node != None:
    lst.append(temp)
    temp = temp.parent_node

for i in lst:
    print(i)


print(x._globalCost)
print(cost)
```