def bfs(src,target):

queue = []

queue.append(src)

exp = []

print("Possible Moves:")

while len(queue) > 0:

source = queue.pop(0)

exp.append(source)

print("\n")

prnt(source)

if source==target:

print("\n")

print("Successfully solved 8 puzzle game!!")

return

poss\_moves\_to\_do = []

poss\_moves\_to\_do = possible\_moves(source,exp)

for move in poss\_moves\_to\_do:

if move not in exp and move not in queue:

queue.append(move)

def prnt(source):

x=0

for i in range(3):

for j in range(3):

print(source[x],end=" ")

x=x+1

print("\n")

def possible\_moves(state,visited\_states):

b = state.index(-1)

d = []

if b not in [0,1,2]:

d.append('u')

if b not in [6,7,8]:

d.append('d')

if b not in [0,3,6]:

d.append('l')

if b not in [2,5,8]:

d.append('r')

pos\_moves\_it\_can = []

for i in d:

pos\_moves\_it\_can.append(gen(state,i,b))

return [move\_it\_can for move\_it\_can in pos\_moves\_it\_can if move\_it\_can not in visited\_states]

def gen(state, m, b):

temp = state.copy()

if m=='d':

temp[b+3],temp[b] = temp[b],temp[b+3]

if m=='u':

temp[b-3],temp[b] = temp[b],temp[b-3]

if m=='l':

temp[b-1],temp[b] = temp[b],temp[b-1]

if m=='r':

temp[b+1],temp[b] = temp[b],temp[b+1]

return temp

src = [0,0,0,0,0,0,0,0,0]

target = [0,0,0,0,0,0,0,0,0]

def inp():

print("Enter input arr:")

for i in range(9):

a=int(input())

src[i]=a

print("Enter target arr:")

for i in range(9):

a=int(input())

target[i]=a

bfs(src, target)

inp()

**OUTPUT :**

Enter input arr:

1

2

3

4

5

6

-1

7

8

Enter target arr:

1

2

3

4

5

6

7

8

-1

Possible Moves:

1 2 3

4 5 6

-1 7 8

1 2 3

-1 5 6

4 7 8

1 2 3

4 5 6

7 -1 8

-1 2 3

1 5 6

4 7 8

1 2 3

5 -1 6

4 7 8

1 2 3

4 -1 6

7 5 8

1 2 3

4 5 6

7 8 -1

Successfully solved 8 puzzle game!!