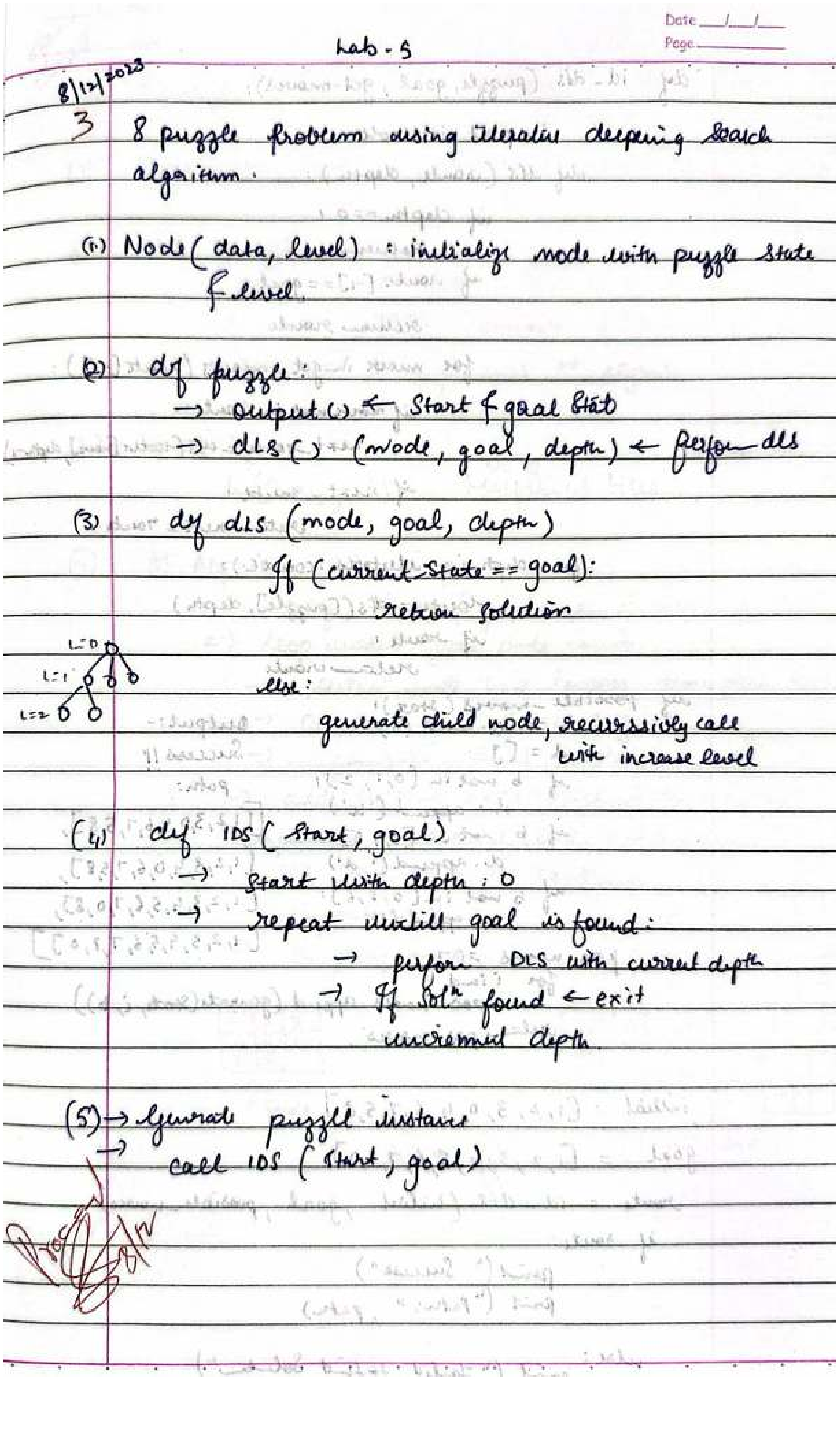
8 puzzle iterative deepening search:-

Code:-



Date __/__/__
Page ____

Lab - 5

8/12/2023

3 8 puzzle problem using iterative deepening search algorithm.

```
(1) Node(data, level) : initialize node with puzzle state
        & level.

(2) def puzzle:
        → output() ← start & goal state
        → dls() ← (node, goal, depth) ← perform dls

(3) def dls (node, goal, depth)
        If (current state == goal):
            return solution
        else:
            generate child node, recursively call
                with increase level

(4) def IDS ( start, goal)
        → start with depth : 0
        → repeat untill goal is found:
            → perform DLS with current depth
            → If soln found ← exit
            incremnt depth.

(5) → generate puzzle instance
    → call IDS (start, goal)
```

[illegible]

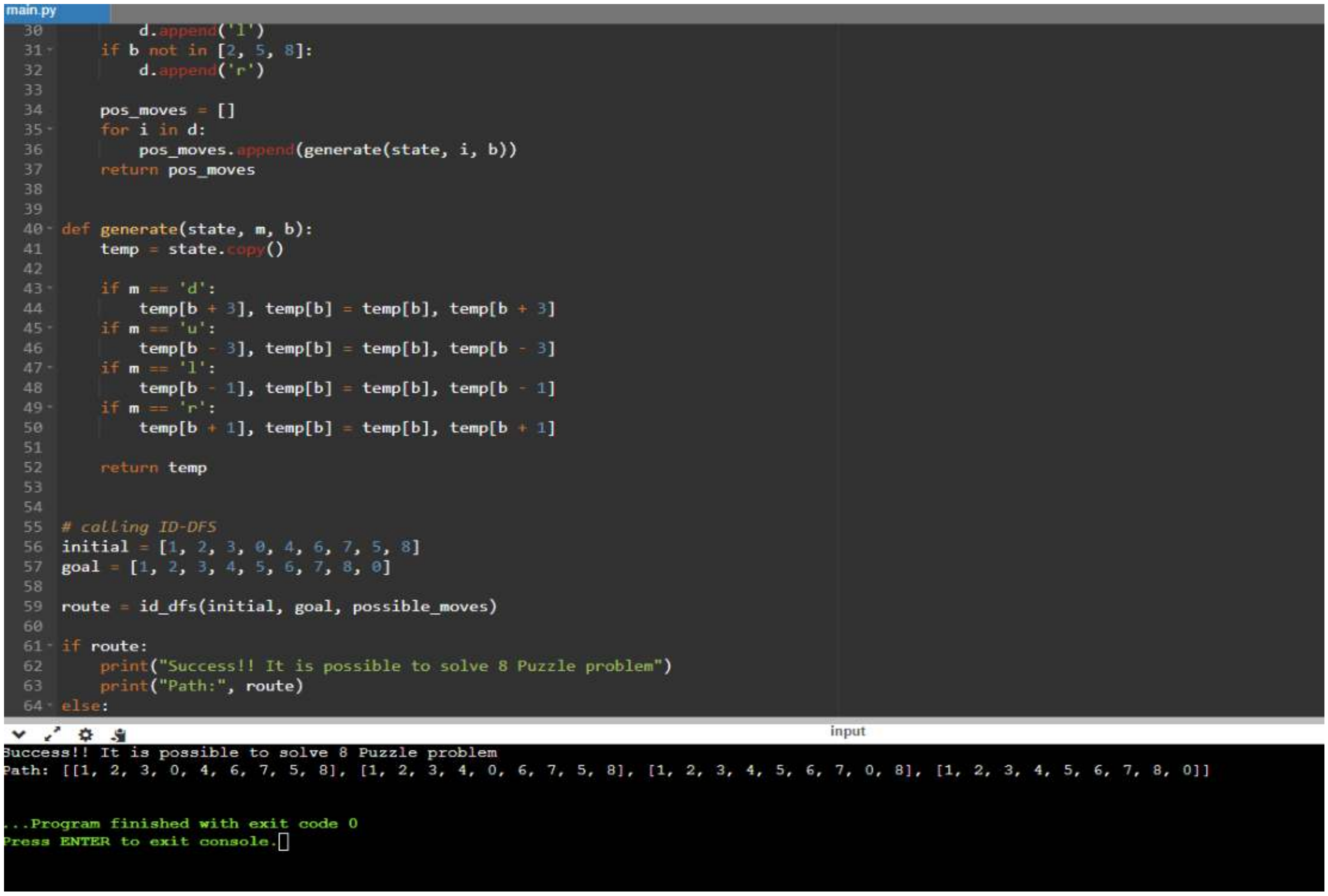
```
def id_dfs (puzzle, goal, get-moves):
        import itertools
        def dfs (route, depth):
            if depth ==0 :
                return
            if route [-1]==goal:
                return route
            for move in get_moves (route[-1]):
                if move not in route:
                    next_route = dfs (route+[move], dep
                    if next_route:
                        return next_route
        for depth in itertools.count ( ):
            route = dfs ([puzzle], depth)
            if route:
                return route

def possible_moves (state):
        b = state.index (0)
        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 (1)
    pos moves =[]
        for i in d :
            pos_moves.append (generate(state, i, b))
        return pos_moves.

initial = [1,2,3,0,4,6,7,5,8]
goal = [1,2,3,4,5,6,7,8,0]
route = id_dfs (initial ,goal , possible_moves)
if route:
        print (" Success")
        print ("Path: " , path)
else:
        print ("Failed to find Solution")
```

```
Output:-
Success !!
path:
[[1,2,3,0,4,6,7,5,8],
[1,2,3,4,0,6,7,5,8],
[1,2,3,4,5,6,7,0,8],
[1,2,3,4,5,6,7,8,0]]
```

output:-



main.py

```
        d.append('l')
    if b not in [2, 5, 8]:
        d.append('r')

    pos_moves = []
    for i in d:
        pos_moves.append(generate(state, i, b))
    return pos_moves


def generate(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


# calling ID-DFS
initial = [1, 2, 3, 0, 4, 6, 7, 5, 8]
goal = [1, 2, 3, 4, 5, 6, 7, 8, 0]

route = id_dfs(initial, goal, possible_moves)

if route:
    print("Success!! It is possible to solve 8 Puzzle problem")
    print("Path:", route)
else:
```

```
Success!! It is possible to solve 8 Puzzle problem
Path: [[1, 2, 3, 0, 4, 6, 7, 5, 8], [1, 2, 3, 4, 0, 6, 7, 5, 8], [1, 2, 3, 4, 5, 6, 7, 0, 8], [1, 2, 3, 4, 5, 6, 7, 8, 0]]


...Program finished with exit code 0
Press ENTER to exit console.
```