LAB 3

=> 8 puzzle program using bfs.

```
def bfs (src, target):

        visited_states = [ ]
        visited_states. append (src)
        arr = [src]
        c = 0
        while arr:
                c+=1
                if arr[0] == target:
                        return True
                arr += possible_moves (arr[0], visited stat)
                arr. pop(0)
        return false

def     possible_moves (state, visited_states):

        b = state. index (-1)
        d = []

        if b+3 in range (9):
                d. append ('d')
        if b-3 in range (9):
                d. append ('u')
        if b not in [0, 3, 6]:
                d. append ('c')
        if b not in [2, 5, 8]:
                d. append ('r')

        pos_moves = []
```

```python
    for moves in d:
        pos_moves.append(gen(state, moves, b))

    return [move for move in pos_moves if
            move not in visited_states]


def gen(state, direction, blank_spot):
    temp = state.copy()

    if direction == 'd':
        a = temp[blank_spot + 3]
        temp[blank_spot + 3] = temp[blank_spot]
        temp[blank_spot] = a

    elif direction == 'u':
        a = temp[blank_spot - 3]
        temp[blank_spot - 3] = temp[blank_spot]
        temp[blank_spot] = a

    elif direction == 'c':
        a = temp[blank_spot - 1]
        temp[blank_spot - 1] = temp[blank_spot]
        temp[blank_spot] = a

    elif direction == 'r':
        a = temp[blank_spot + 1]
        temp[blank_spot + 1] = temp[blank_spot]
        temp[blank_spot] = a

    return temp
```

```
# Test 1
src = [1,2,3,-1,4,5,6,7,8]
target = [1,2,3,4,5,-1,6,7,8]
print (bfs (src, target))
```