Nidhish Lakhinana
1BM18CS079

8 Puzzle 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_states)
        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 rang(9);
        d.apend ('u');
    if b not in r(9):
        d. apnd ('l'):
```

```
    if b not in [a, r, 8]:
        append ('r')

poss_moves = []

for move in d:
    pos_moves.append (gen(stak, move, b))
return [move for move in pos_moves if move not in visited stak]


def gn (stak, direct, blan spot):
  temp = stak.copy()
  if direch == 'd':
      a = temp [bs + 3]
      temp [bs + 3] = temp [bs]
      temp [bs] = a

  ~~if direct == 'u':~~
  if dirct == 'u':
      a = temp [bs - 1]
      temp [bs - 1] = temp [bs]
      temp [bs] = a

  if dir == 'l'
      a = tem [bs - 1]
      tem [bs - 1] = tem [bs]
      temp [bs] = a

  if dir = 'r':
      a = tmp [bs + 1]
      tem [bs + 1] = tem
      ~~temp = a~~
  return temp
```