M.S. SATHWICK KIRAN

1BM18CS050

⇒ 8 PUZZLE PROGRAM :-

```
def dfs(src, target, limit, visited_states);
if src == target;
    return true
  if limit <= 0
      return False
 visited states. append(src)
 adj = possible moves (src, visited_states)
 for move in adj:
       if dfs(move, target, limit -1, visited_states):
           return true
     return False
  def possible_moves (state, visited states):
   ind = state.. index (-1)
  d= []
  if ind+3 in range (9):
      d. append ('d')
     if ind-3 in range (9):
         d. append ('u')
    if ind not in [0, 3, 6]:
        d: append ('l')
   if ind not in [2, 5, 8]:
        d: append ('r')
```

```
=>  pos_moves = []

    for mov in d :
        pos_moves.append(gen(state, move, ind))
      return [mov for move in pos-moves if move not in visited_state]
def gen ( state , m , b):
      temp = state . copy ()
    if m = = 'd';
       a = temp [b+3]
       temp [b+3] = temp [b]
       temp [b] = a
    elif  m = 'n':
          a = temp [b-3]
          temp [b-3] = temp [b]
          temp [b] = a
    elif  m == 'L':
          a = temp [b-1]
          temp [b-1] = temp [b]
          temp [b] = a
       elif  m == 'r':
             a = temp [b+1]
             temp [b+1] = temp b
             temp [b] = a
    return temp
```

```
def iddfs (src, target , depth):
    visited_ state []

    for i in range (1, depth +1):
        if dfs (src , target , i, visited_ state):
            return True
    return False.
```