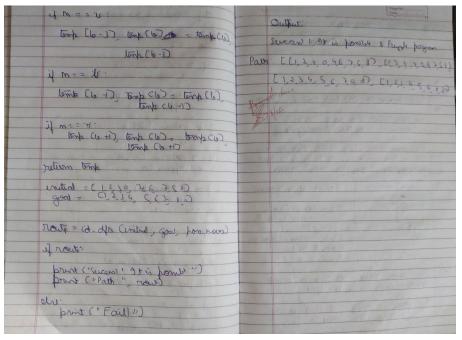
Program 4 Catalacian relinad fits 8 purole using Iteration Deeper le = state index (0) def des (rout apth): (c,1,0) is loss of fi d append ('u') if depth = = 0: if b not in [678): return dappend (a') if Prout [-D = = you! il 6 not in [03,67: return rout dappend ('l') don more in get mores (mout(-1)): 4 6 not in (2,53): dappend (5) I move not in rout: pos Morses = () nest nout = aps (nout + thous depth +) don ind if host nout ! pos mas append (gonerati (state @; 6)) relian next rout return por nous for depth in itertoob count() det gennate (stat, m, o). rout = als (chunge), depth ) temp = state copy() if noute: if m = = d .: return rout temp ( 6 +3) temp (6) = temp(0) temp (6+3)



(2) Harshala Rani-1bm21cs074
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]]