**A\* Algorithm**

**Aim :** Create a program to depict the working of astar algorithm.

**8 Puzzle Problem**

goal=[[0,1,2],[3,4,5],[6,7,8]]

dummy=[[0,0,0],[0,0,0],[0,0,0]]

g=0

f=[999]\*4

hprev=999

prevind=-1

start=list()

print("\nEnter the initial state")

for i in range(3):

start.append([int(x) for x in input().split()])

def check(intermediate):

hfinal=0

for i in range(3):

for j in range(3):

if goal[i][j]==intermediate[i][j] :

pass

else:

if(intermediate[i][j]==0):

pass

else:

hfinal=hfinal+1

return hfinal

hfinal=check(start)

#h[up,down,left,right]

def iterate():

global g

g=g+1

global prevind

h=[999]\*4

for i in range(3):

if 0 in start[i]:

j=start[i].index(0)

if i==0:

h[1]=down(i,j)

move(i,j,h)

ind=h.index(min(h))

prevind=previous(h,i,j,ind)

elif i==1:

h[0]=up(i,j)

h[1]=down(i,j)

move(i,j,h)

ind=h.index(min(h))

prevind=previous(h,i,j,ind)

elif i==2:

h[0]=up(i,j)

move(i,j,h)

ind=h.index(min(h))

prevind=previous(h,i,j,ind)

#shift(ind,i,j)

else:

continue

break

def copy(dummy,start):

for i in range(3):

for j in range(3):

dummy[i][j]=start[i][j]

def previous(h,i,j,ind):

global hprev

if (prevind==0 and ind==1) or (prevind==1 and ind==0) or (prevind==2 and ind==3) or (prevind==3 and ind==2):

h[ind]=999

ind=h.index(min(h))

#if(hprev>=min(h)):

#hprev=min(h)

return shift(i,j,ind)

#else:

#return 999

def shift(i,j,ind):

if ind==0:

up(i,j)

copy(start,dummy)

print("shifted up")

display(start)

return 0

elif ind==1:

down(i,j)

copy(start,dummy)

print("shifted down")

display(start)

return 1

elif ind==2:

left(i,j)

copy(start,dummy)

print("shifted left")

display(start)

return 2

else:

right(i,j)

copy(start,dummy)

print("shifted right")

display(start)

return 3

def move(i,j,h):

if j==0:

h[3]=right(i,j)

elif j==1:

h[3]=right(i,j)

h[2]=left(i,j)

else:

h[2]=left(i,j)

def up(i,j):

copy(dummy,start)

dummy[i][j],dummy[i-1][j]=dummy[i-1][j],dummy[i][j]

return check(dummy)

def down(i,j):

copy(dummy,start)

dummy[i][j],dummy[i+1][j]=dummy[i+1][j],dummy[i][j]

return check(dummy)

def left(i,j):

copy(dummy,start)

dummy[i][j],dummy[i][j-1]=dummy[i][j-1],dummy[i][j]

return check(dummy)

def right(i,j):

copy(dummy,start)

dummy[i][j],dummy[i][j+1]=dummy[i][j+1],dummy[i][j]

return check(dummy)

def display(intermediate):

for i in range(3):

for j in range(3):

print(intermediate[i][j],end=" ")

print("\r")

print("\nGoal State")

display(goal)

while(goal!=start):

if(prevind==999):

print("\nLocal Minima at cost : ",g)

break

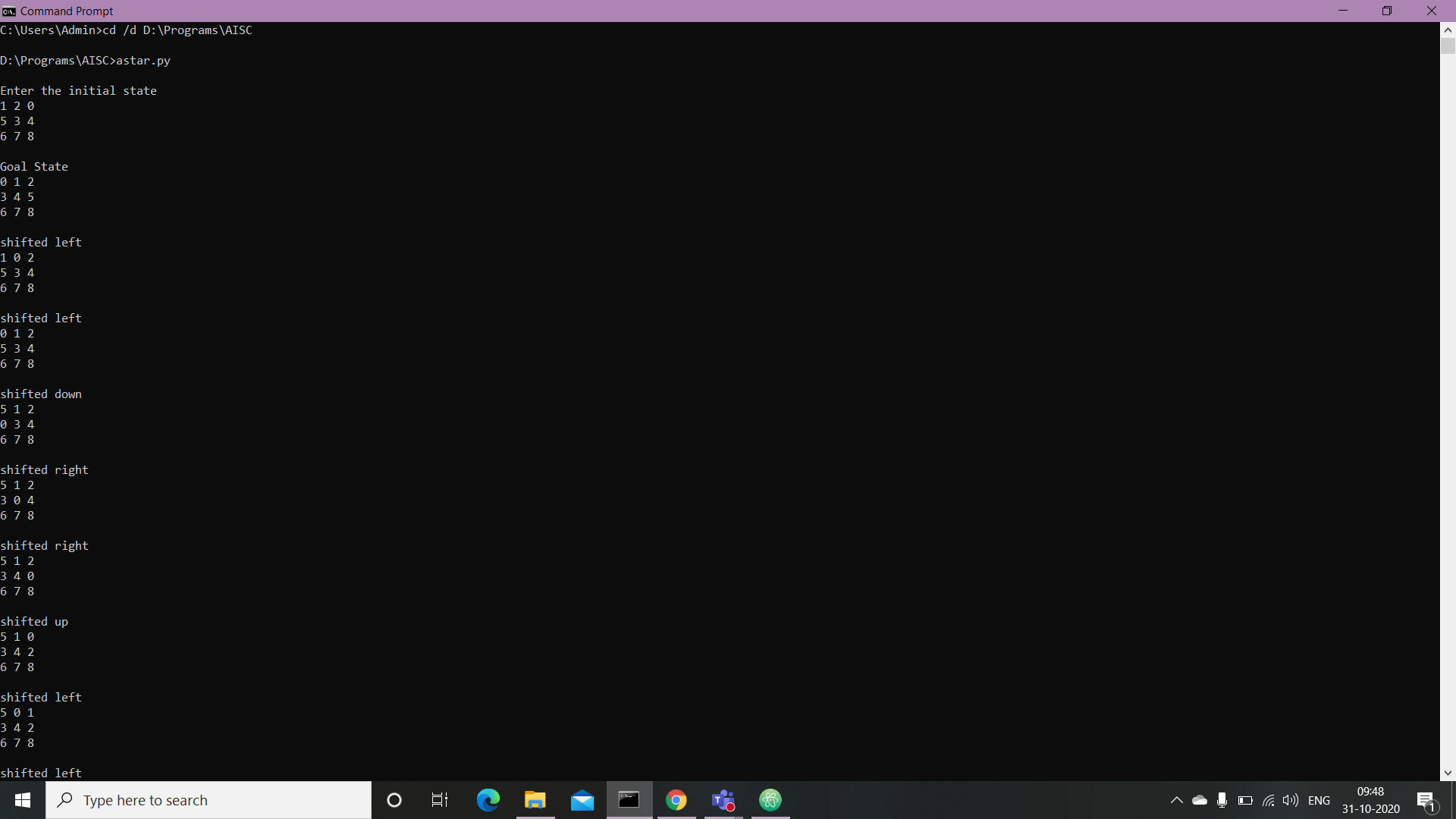
print()

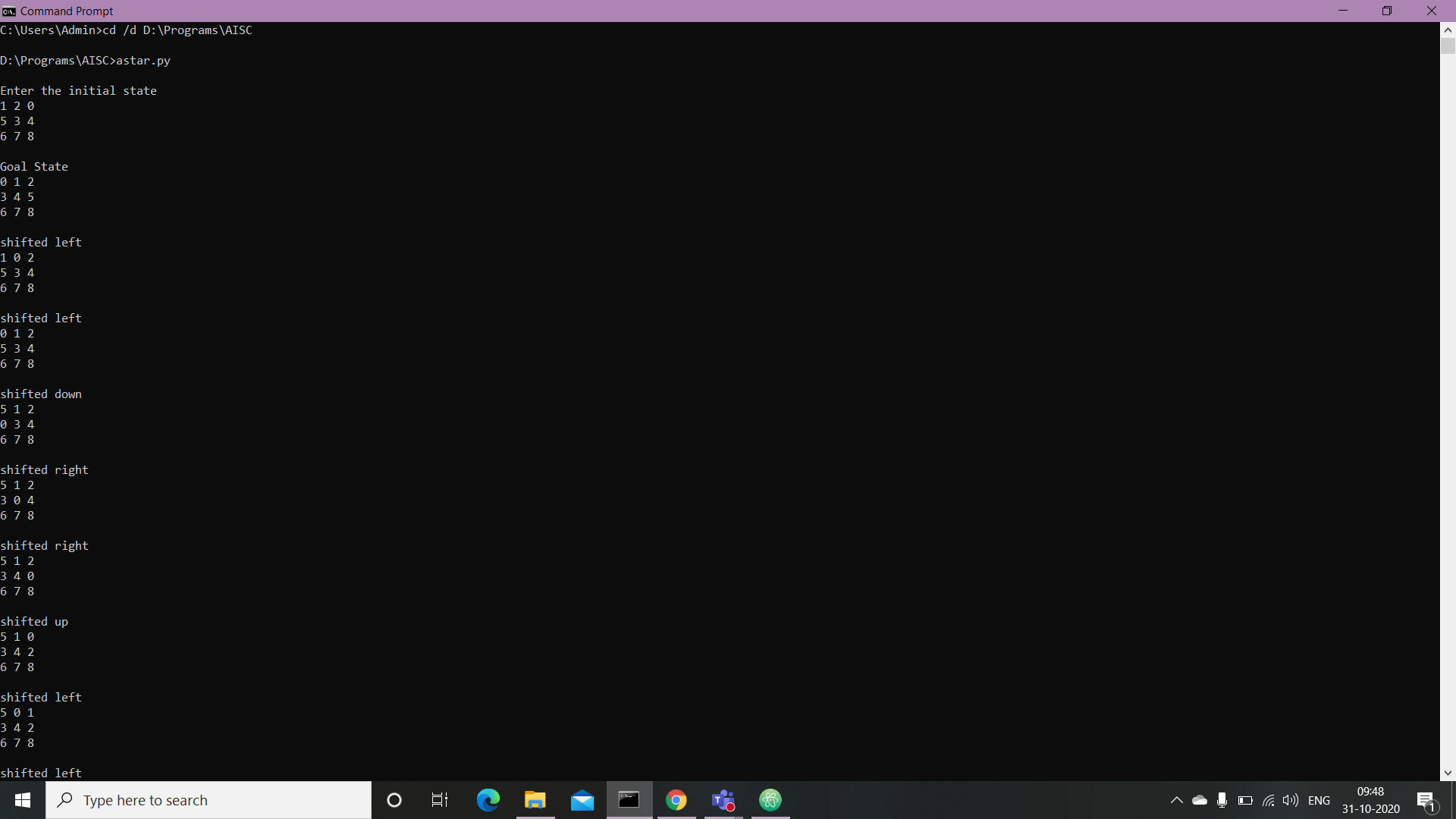
iterate()

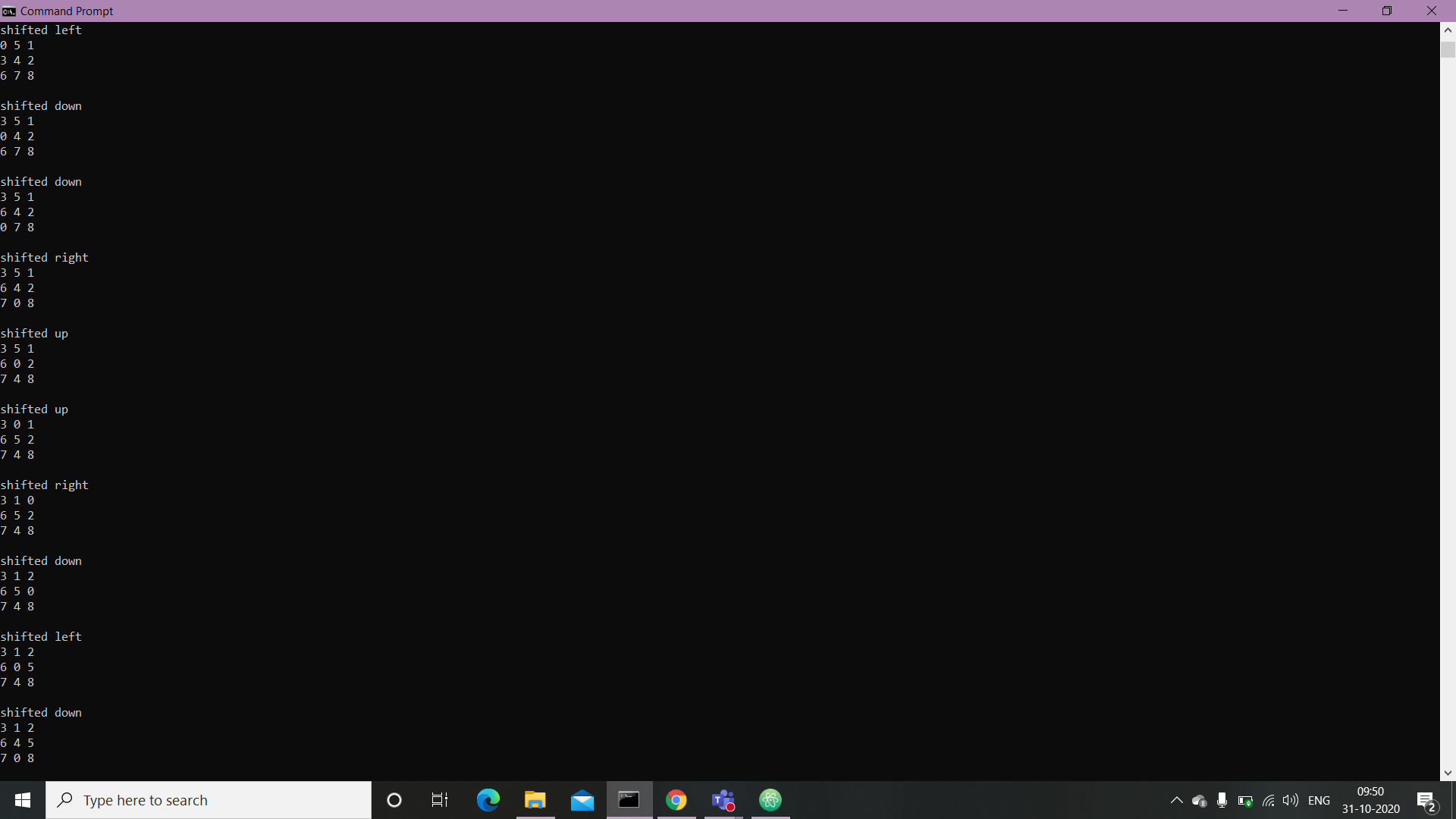
if(prevind!=999):

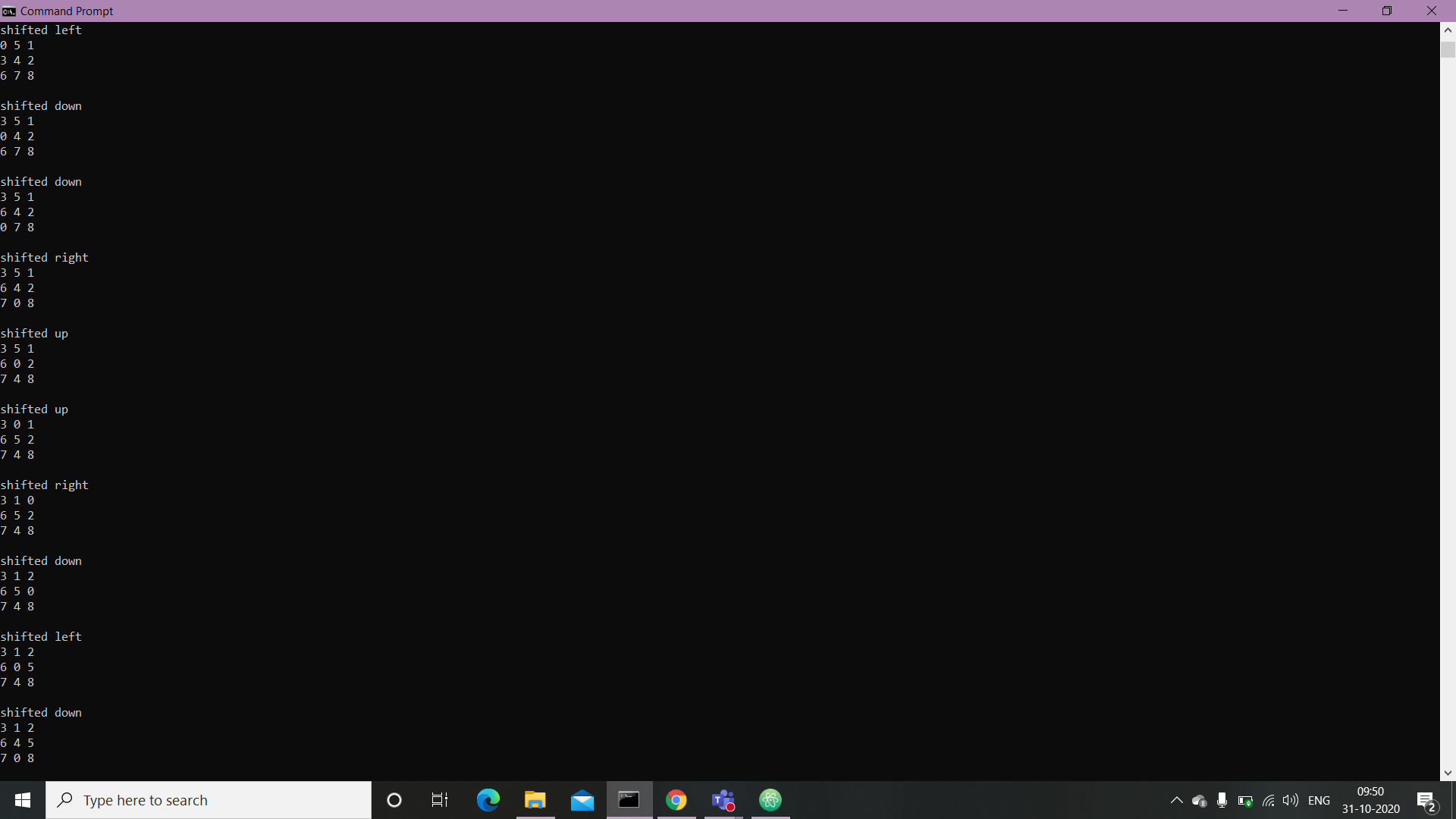
print("\nMatrix is in goal state, cost : ",g)

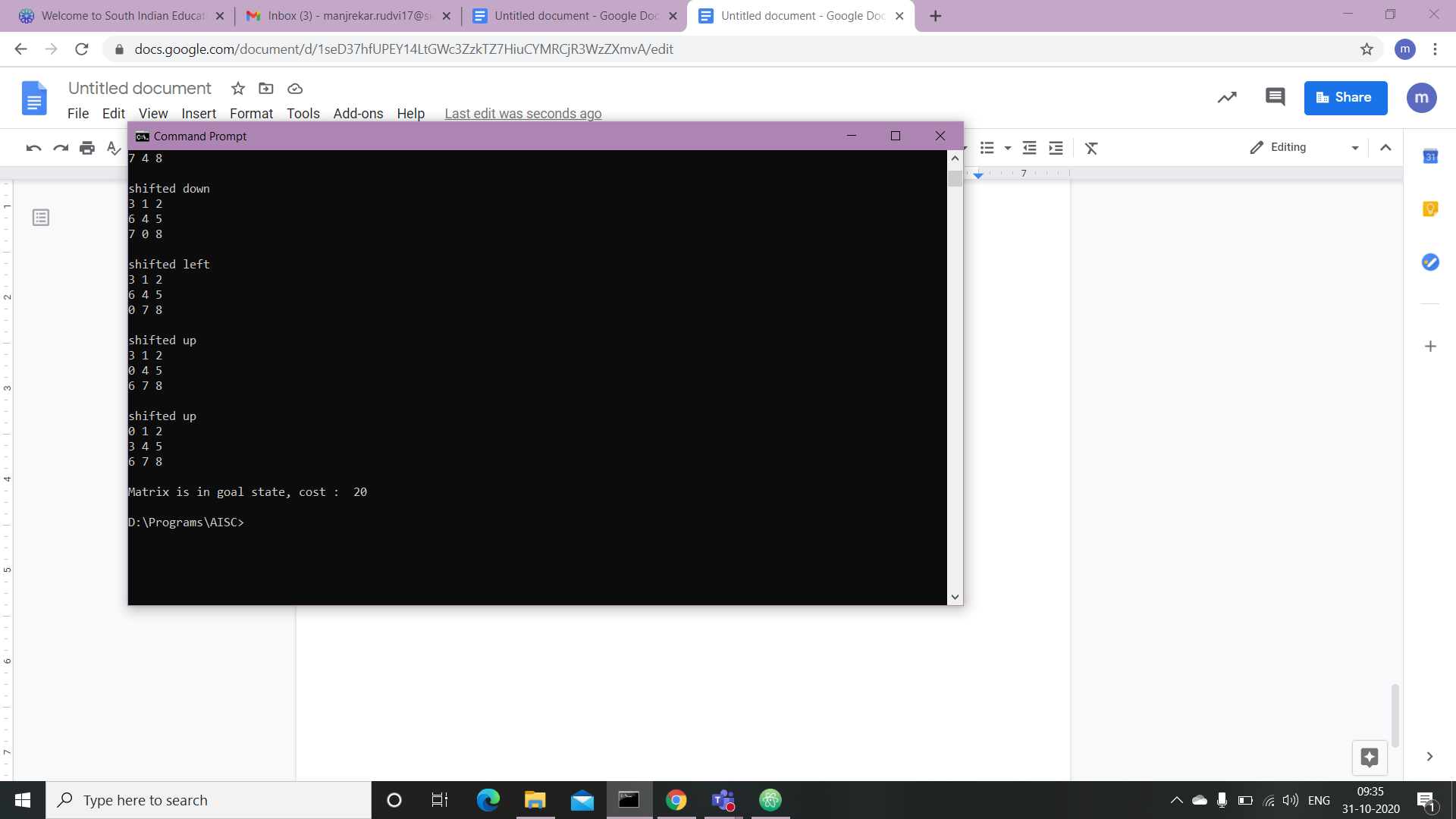
**Output :**

****

****

****

****

****

**Conclusion :** Hence, we have successfully designed 8 Puzzle Problem using A\* algorithm.