5th sem

AI LAB TEST 1

SOWMYA.D.V.
1BM19CS408

program - 1

```
class Node:
def _init_ (self, data, level, fval):
"""Initialize the node with the data, level of
  the node & the calculated fvalue""

  self.data =data
  self.level =level
  self.fval =fval

def generate_child(self):

 x ,y = self.find (self.data, '_')
 val_list = [[x,y-1], [x ,y+1], [x-1,y], [x+1,y]]
 children=[]

for i in val_list :
 child= self.shuffle (self.data, x,y, i[0], i[1])
    if child is not None:
 child_node = Node (child, self.level+1, 0)
   children.append (child_node)
  return children.

def shuffle (self, puz, x1, y1, x2, y2 ):
 if x2 >= 0 and x2 < len (self.data) and y2 >=0
 and y2 < len (self.data):
      temp_puz = []
      temp_puz = self.copy (puz)
      temp = temp_puz [x2] [y2]
      temp_puz [x2][y2] = temp_puz [x1] [y1]
      temp_puz [x1] [y1] = temp
      return temp_puz
  else:
      return None
```

Sowmya.D.V.

SOWMYA D.V.
1BM19CS408

```
def copy (self, root):
    temp =[]
    for i in root:
        t =[]
        for j in i:
            t.append (j)
        temp.append (t)
    return temp

def find (self, puz,x):
for i in range (0, len (self.data)):
    for j in range (0, len (self.data)):
        if puz [i] [j] ==x:
            return i, j

class Puzzle:
def _init_ (self, size):
    self.n=size
    self.open =[]
    self.closed =[]
def accept (self):
    puz =[]
    for i in range (0, self.n):
        temp =input(). split ("  ")
        puz.append (temp)
    return puz.
def f (self, start, goal):
    return self.h (start.data, goal)+ start.level
```

[illegible]

SOWMYA.D.U.
1BM19CS408

```
def h (self , start , goal):
  temp = 0
for i in range (0, self.n):
   for j in range (0, self.n):
     if start [i][j] != goal [i][j] and start [i][j]
              != '_':
    temp + = 1
return temp.
```

[illegible]