0/11/20

AI-LAB TEST

Abhishikat Soni
1BM18CS004
5A

```
import math                 #Maze Problem
maze=[]                     #A*
path=[]
closedPath=[]
neighbours=[[1,1],[0,1],[1,0],[1,-1],[0,-1],[-1,1],[-1,0],
            [-1,-1]]

def euclid(x,n,m):
    dist=math.sqrt((n-1-x[0])*2+(m-1-x[1]*2)
    return dist

def findShortestPath(nextPath,n,m):
    minDistance = 999
    next=[]
    for x in nextPath:
        if(euclidDist(x,n,m) < minDistance)
            minDistance = euclidDistance(x,n,m)
            next=x
    return next

def findPath(n,m):
    path.append([0,0])
    current=[0,0]
    while(current != [n-1,m-1]):
        nextPath=[]
        for x in neighbours:
         a=[]
         a.append(current[0]+x[0])
         a.append(current[1]+x[1])
         if a[0]>-1 and a[0]<n and a[1]>-1 and a[1]<m:
         #  if(maze[a[0]][a[1]]:
                if a not in path and a not in closedPath:
                   nextPath.append(a)
        if(nextPath):
            current = findShortestPath (nextPath, n, m)
            Path.append (Current)
      else:
          if path:
              closedPath.append (Current)
              Path.pop()
              if path:
                  Current = path[len(path)-1]
              else:
                  print ("NO PATH")
        else:
            print ("NO PATH")
            exit(0)

def start():
    n = int(input("Row"))
    m = int(input("Enter Column"))
    print("Enter maze structure (0-blocked, 1-free):")
    for i in range(n):
        a=[]
        a=list(map(int, input().split(" ")))
        maze.append(a)
    print("Maze\n")
    for i in range(n):
        for j in range(m):
            if([i, j] in path);
                print("_ ", end=" ")
            else:
                print(maze[i][j], end=" ")
        print()
    print()
print (parm)
~~if~~
if _name_ == "_main_" :
    start()
```