# **LAB 1 - TOY PROBLEM**

RA1811003010513

Aditya Jain

**Problem :** Given a string path, where path[i] = 'N', 'S', 'E' or 'W', each representing moving one unit north, south, east, or west, respectively. You start at the origin (0, 0) on a 2D plane and walk on the path specified by path.

Return True if the path crosses itself at any point, that is, if at any time you are on a location you've previously visited. Return False otherwise.

**Solution (C++) :**

x = y = 0

s = {(0, 0)}

for p in path:

if p == "N":

y += 1

elif p == "S":

y -= 1

elif p == "W":

x += 1

else:

x -= 1

if (x, y) in s:

return True

s.add((x, y))

return False

Input 1 : "NESWW"

Output 1 : True

Input 2 : "NES"

Output 2 : False

1.What is a toy problem?

Ans : A toy problem is a problem that doesn't have all the complexity of a real-world engineering problem. It could be a simplified or shallow version of a more difficult and intricate problem or class of problems.

Solving a small toy problem can be beneficial in the context of learning a new field or skill, as an exercise, or as a way to eventually work your way into the larger, real-world problem.

2.How many approaches do you have for solving the toy problem which you have taken?

Ans : There are multiple approaches, the above one is using a set to store previously visited points. Simulate the process while keeping track of visited points. It uses if-else statements to compare and proceed further.

3.Map your toy problem to any real life application.

Ans : Navigational Instruments might uses this to phenomenon while trekking in forest or on mountains. Automobile industries are using this technology in GPS tracking used in cars.