**qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm**

|  |
| --- |
| Modeling and Simulation  Assignment: Pure Pursuit Problem  5/25/2021  **Submitted To: Dr. S.M.K. Quadri**  Sub. By : Sandeep Bhatt, Roll no: 20MCA051 |

**Q1. Implement Pure Pursuit Problem in any language you have been taught during your MCA program.**

1. **Code of the program**

#!/usr/bin/env python3

import math

def pure\_pursuit\_simulation(x\_fighter, y\_fighter, x\_bomber, y\_bomber, VELOCITY\_FIGHTER, MAX\_TIME, MAX\_RANGE):

t = 0

while t < 12:

distance = math.sqrt(((x\_bomber[t] - x\_fighter[t])\*\*2) + ((y\_bomber[t] - y\_fighter[t])\*\*2))

print('x\_f :', '{:.2f}'.format(x\_fighter[t]), ',y\_f :', '{:.2f}'.format(y\_fighter[t]), end = "")

print('\tx\_b :', '{:.2f}'.format(x\_bomber[t]), ',y\_b :', '{:.2f}'.format(y\_bomber[t]), end = "")

print('\tDistance :', '{:.2f}'.format(distance),'KMS', end = "")

print('\tTime : ', t)

if distance <= MAX\_RANGE:

break;

x\_fighter[t + 1] = x\_fighter[t] + VELOCITY\_FIGHTER \* ((x\_bomber[t] - x\_fighter[t]) / distance)

y\_fighter[t + 1] = y\_fighter[t] + VELOCITY\_FIGHTER \* ((y\_bomber[t] - y\_fighter[t]) / distance)

t += 1

if t < 12:

print('\nBomber Destroyed At Time =', t, ', Distance= ', '{:.2f}'.format(distance), 'kms')

else:

print('\nBomber Escaped...')

MAX\_TIME = 12 # MAX Time To Attack Bomber

MAX\_RANGE = 10 # MAX Distance Fighter Can Attack

VELOCITY\_FIGHTER = 20 # Velocity of Fighter

# Bomber path

x\_bomber = [80, 90, 99, 108, 116, 125, 133, 141, 151, 160, 169, 179, 180]

y\_bomber = [0, -2, -5, -9, -15, -18, -23, -29, -28, -25, -21,-20, -17]

# Initial Position of Fighter

x\_fighter = [0] \* 12

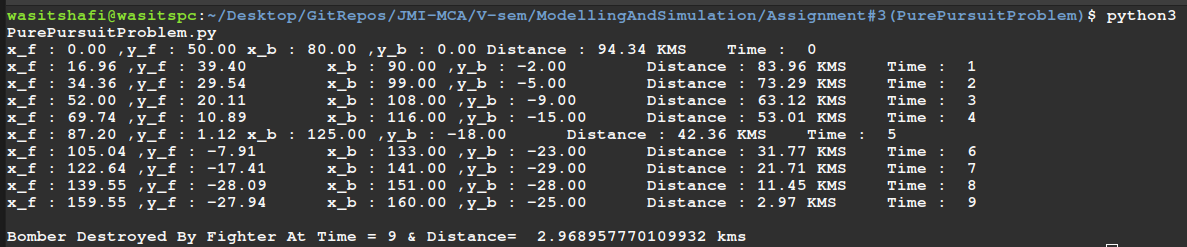
y\_fighter = [0] \* 12

x\_fighter[0] = 0

y\_fighter[0] = 50

pure\_pursuit\_simulation(x\_fighter, y\_fighter, x\_bomber, y\_bomber, VELOCITY\_FIGHTER, MAX\_TIME, MAX\_RANGE)

1. **Results**



1. **Discussion, if any**

* Pure pursuit is a type of pursuit curve used in aerial combat in which an aircraft pursues another aircraft by pointing its nose directly towards it.
* Pure Pursuit: When target is not aware of pursuer. In this case the course of target is known.
* Hot Pursuit: When target is aware of the pursuer.
* A fighter aircraft sights an enemy bomber and flies directly towards it in order to catch up the bomber and destroys it.
* The bomber continues flying so that the fighter has to change its direction to keep pointed towards the target.
* If the target flies along a straight, the problem can be solved directly with analytical techniques.

**We are given following conditions:**

1. Both target and pursuer are flying in the same 2 dimensional plane.

2. The fighter's speed is constant that is VELOCITY\_FIGHTER.

3. The target's path is known.

4. Minimum distance required by the fighter to fire a missile at bomber is 10 units.

5. If the target is not caught within given time t (here t = 12), the target(bomber) escapes.

6. Initial coordinates of the pursuer (fighter) are known.

**On Sumulating we found fighter hits Missile at bomber at time = 9, and distance = 2.96kms**