def onSegment(p,q,r) :

if (q.x <= max(p[0], r[0]) and q[0] >= min(p[0], r[0]) and q[1] <= max(p[1], r[1]) and q[1] >= min(p[1], r[1])):

return True

return False

def orientation(p,q,r):

val = (q[1] - p[1]) \* (r[0] - q[0]) - (q[0] - p[0]) \* (r[1] - q[1])

if (val == 0):

return 0

else:

if (val >0 ):

return 1

else:

return 2

def doIntersect(p1,q1,p2,q2):

o1 = orientation(p1, q1, p2)

o2 = orientation(p1, q1, q2)

o3 = orientation(p2, q2, p1)

o4 = orientation(p2, q2, q1)

temp = onSegment(p1, p2, q1)

if (o1 != o2 and o3 != o4):

return True;

if (o1 == 0 and temp):

return True

if (o2 == 0 and temp):

return True

if (o3 == 0 and temp):

return True

if (o4 == 0 and temp):

return True

return False

def IsInside(polygon,sides,point):

if (sides < 3):

return False;

extreme = (1000000,point[1]);

count=0

i=0

while True:

next = (i+1)%sides

if (doIntersect(polygon[i], polygon[next],point, extreme)):

if (orientation(polygon[i],point, polygon[next]) == 0):

return (onSegment(polygon[i],point, polygon[next]))

count=count+1

i = next

if (i==0):

break

return (count%2 == 1)

polygon= [(0, 0), (10, 0), (10, 10), (0, 10)]

sides= len(polygon)

point=(10,10)

print(point)

result= IsInside(polygon,sides,point)

if result=='True':

print("Yes")

else:

print("No")