```
#Jack Amos
#CS2300-002
#10/04/2019
#Project 2
#using Python 3.7
import random
def print_coordinates(filename):
       with open(filename,'r') as file:
              contents = file.read()
       #get each value in string an convert it to integer
       # [llx,lly,d,p1,p2,v1,v2]
       coord_values = [int(x) for x in contents.split()]
       #sets scale of "screen"
       dimension = 41
       #2d array that will be printed to fucntion as screen, random numbers exist so python does not
make all indexes of non-unique elements 0
       screen = [[str(random.randint(0,10000))] for x in range(41)] for x in range(41)]
       implicit = get implicit(coord values)
       #figure out points
       if coord_values[0] > 0:
              x_point = 20 - coord_values[0]
       elif coord_values[0] <= 0:</pre>
              x_point = 20 + coord_values[0]
       y_point = 20 - coord_values[1]
       r1 = range((y_point-coord_values[2]+1),y_point+1)
       index = x_point
       #make actual box
       for n in screen:
              if screen.index(n) in r1:
                      n[x_point] = "*"
                      try:
                             n[x_point+coord_values[2]] = "*"
                      except:
                             pass
              if screen.index(n) == y\_point or screen.index(n) == (y\_point-coord\_values[2]+1):
                      while index < (x_point+coord_values[2]):
                             try:
                                     n[index] = "*"
                             except:
```

```
index+=1
                      index = x\_point
       #draw line
       for n in screen:
               for i in n:
                       if (implicit[0]*(20-screen.index(n))+implicit[1]*(n.index(i)-20)+implicit[2]) ==
0:
                              screen[screen.index(n)][n.index(i)] = "*"
       y_axis = 20
       #add axis
       for n in screen:
               if y_axis < 0 and y_axis > -10 or y_axis > 9:
                      n.insert(0,str(y_axis)+" ")
               elif y axis > -1:
                      n.insert(0,str(y_axis)+" ")
               else:
                      n.insert(0,str(y_axis))
               y_axis-=1
       #removes random numbers but not axis
       for n in screen:
               for i in n:
                      if i.isdigit() and n.index(i) != 0:
                              n[n.index(i)] = " "
       row = 0
       screen_string = ""
       while row < dimension:
               screen_string += str(screen[row]) + "\n"
               #increment by column number so vars are at the start and end of next row
               row += 1
       #removes unneeded chars for beautification
       screen_string = screen_string.replace(",","").replace("[","").replace("]","").replace(""","")
       #print final graph
       print(screen_string)
       print("-20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20")
       print("\n")
```

pass

```
print("Implicit Form: "+str(implicit[0])+"x+"+str(implicit[1])+"x+"+str(implicit[2])+"=0")
print("_____
                                                        _\n")
#gets values for implicit equation
def get_implicit(coord_values):
       #[a,b,c]
      implicit_values = []
       implicit_values.append(-1*coord_values[5])
       implicit_values.append(coord_values[6])
      implicit_values.append(-1*(coord_values[6]*coord_values[3]+(-
1*coord_values[5]*coord_values[4])))
       return implicit_values
#call func
print_coordinates("line1-1.txt")
print_coordinates("line2.txt")
print_coordinates("line3.txt")
print_coordinates("line4.txt")
```

print_coordinates("line5.txt")