

```

import sys
import io
import itertools

#Problem 1

txt = """6
5 19
55 28
38 101
28 62
111 84
43 116
5
11 27
84 99
142 81
88 30
95 38
3
132 73
49 86
72 111
0"""

stdin = io.StringIO(txt)

#Actual use (Comment the below line for testing)
stdin = sys.stdin

def find_distance(p1, p2):
    return ((p1[0]-p2[0])**2 + (p1[1] - p2[1])**2)**(0.5)

def find_cable_network(computers, positions, n):
    p = set(positions)
    min_distance = 100000000
    x = 0
    for item in itertools.product(p, repeat=computers):

        flag = 0
        for c in item:
            if item.count(c) > 1:
                flag = 1
        if not flag:
            network = list(item)
            total_distance = 0
            stretches = []
            for i in range(computers - 1):
                distance = find_distance(item[i], item[i+1])
                stretches.append(distance)
                total_distance += distance

            if total_distance < min_distance:
                min_distance = total_distance
                opt_network = network
                opt_stretches = stretches

    return (opt_network, opt_stretches)

```

```

def P1():
    count = 1    #keps count of networks
    while(True):
        computers = int(stdin.readline().strip())    #Number of computers
        if not computers:
            return
        print("*****")
        print(f"Network #{count}")
        positions = []    #Store computers
        for _ in range(computers):
            p = list(map(int, stdin.readline().split()))
            p = (p[0], p[1])
            positions.append(p)

        network = find_cable_network(computers, positions, count)
        total_distance = 0
        for i in range(computers - 1):
            total_distance += network[1][i] + 16
            print(f"Cable requirement to connect {network[0][i]} to {network[0]
[i+1]} is {network[1][i]:.2f} feet.")
            print("Number of feet of cable required is {:.2f}
feet.".format(total_distance))
            count += 1

P1()

```